A Method of Integrating Aeroheating into Conceptual Reusable Launch Vehicle Design

Final Report

NCC 2-5332 Closed
NASA Ames Research Center

For a Joint Research Interchange (JRI) Cooperative Agreement entitled

Evaluation of Advanced Thermal Protection Techniques for Future Reusable Launch Vehicles

Submitted by:

Dr. John R. Olds (PI)
Lt. Kris Cowart (thesis author)
Georgia Institute of Technology
Atlanta, GA 30332-0150

CASI
cc: 202A-3

June 2001

ACKNOWLEDGEMENTS

Many people deserve recognition for making this Thesis possible. I will start by thanking God, who has made me realize that no mountain is too big with him on my side, and I want to thank the members of my family for their love and support over the years.

Also, I would like to extend my gratitude to all those people in the Air Force that helped me pursue the Air Force Institute of Technology's Scholarships/Fellowships and Grants Program. These include Col Paul L. Jacobs (Ret), and Col Walter R. Dill (Ret).

I want to also thank the personnel of AFROTC Detachment 165 at Georgia Tech. The many times that I had questions about the Air Force, the Cadre and Staff of DET 165 always pulled through for me.

Dr. Olds, I want to thank you as well for all the help and guidance that you have provided during my stay at Tech. Thanks for taking me under your wing and teaching me the ins-and-outs of RLV conceptual design.

Also, I would like to thank the members of my committee, Dr. Stephen Ruffin and Paul Kolodziej, for the advice and guidance they provided in order to make this thesis possible.

In addition, I would like to thank Reggie Alexander at NASA Marshal for his continued advice about MINIVER and for running several benchmark solutions on SINDA.

A hearty thanks goes out to my co-workers in the Space System Design Lab that have put up with me and have helped me over the past two years. You guys are the best, and it has been great being your friend and working with you all.

This last thank you goes out to my wife and best friend, Nicole. You have made this all possible by being there for me in order to help keep me focused on the important things. I truly do not know what I would do without you at times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ADS	Automated Design Synthesis
AETB-8	8 lbm/ft^3 Alumina Enhanced Thermal Barrier composite tile
AETB-12	12 lbm/ft^3 Alumina Enhanced Thermal Barrier composite tile
AFRSI	Advanced Flexible Reusable Surface Insulation
AFSRI-2200	2200°F temperature limit AFRSI
AFRSI-2500	2500°F temperature limit AFRSI
CFBI	Composite Flexible Blanket Insulation
CFD	Computational Fluid Dynamics
CGI	Common Gateway Interface
CL	Centerline
DSM	Design Structure Matrix
DURAFRSI	Durable AFRSI
FRCI	Fibrous Refractory Composite Insulation
FRSI	Felt Reusable Surface Insulation
GrEx	Graphite Epoxy
GIF	Graphic Interchange Format
HTML	Hypertext Mark Up Language
LI-900	9 lbm/ft^3 rigid ceramic tile insulation
LI-2200	22 lbm/ft^3 rigid ceramic tile insulation
NASA	National Aeronautics and Space Administration
OFT	Orbiter Flight Test
PC	Personal Computer
PBI	Polybenzimidazole Blanket Insulation
RCC	Reinforced Carbon Carbon

RLV	Reusable Launch Vehicle
RSI	Reusable Surface Insulation
SiC	Silicon Carbide
STS	Space Transportation System
TABI	Tailorable Advanced Blanket Insulation
TCAT	Thermal Calculation Analysis Tool
TiAl	Titanium Aluminide
TMM	Thermal Math Model
TPS	Thermal Protection System
TPS*X*	Thermal Protection System Expert
TUFI	Toughened Uni-Piece Fibrous Insulation
U.W.	Unit Weight
WWW	World Wide Web

LIST OF SYMBOLS

$\|\|g\|\|_2$	Vector two norm where g is a vector with N number of elements
c_p	Material specific heat, $\left(\frac{\text{kJ}}{\text{kg K}}\right)or\left(\frac{\text{Btu}}{\text{lbm R}}\right)$
$\frac{dT}{dx}$	Derivative of temperature with respect to x
$\frac{d^2T}{dx^2}$	Second derivative of temperature with respect to x
f_i	System of equations formed after discretization of the heat equation
i	Spatial position: 1,..,N
$J_{i,j}$	Jacobian matrix: i, j = 1,..,N
k	Material thermal conductivity, $\left(\frac{\text{W}}{\text{m K}}\right)or\left(\frac{\text{Btu}}{\text{lbm s R}}\right)$
L	Material stack thickness, (m) or (ft)
N	Maximum number of nodes
n	Time level
q_{cond}	Conductive heat rate, $\left(\frac{\text{W}}{\text{m}^2}\right)or\left(\frac{\text{Btu}}{\text{ft}^2\text{ s}}\right)$
q_{conv}	Convective heat rate, $\left(\frac{\text{W}}{\text{m}^2}\right)or\left(\frac{\text{Btu}}{\text{ft}^2\text{ s}}\right)$
q_{rad}	Radiative heat rate, $\left(\frac{\text{W}}{\text{m}^2}\right)or\left(\frac{\text{Btu}}{\text{ft}^2\text{ s}}\right)$
R_n	Nose radius, (m) or (ft)
S	Running length, (m) or (ft)
T	Temperature, (K)or(F)

T_s	Surface temperature, (K) or (F)
t	Time, (s)
x	Position in material normal to exposed surface, (m) or (ft)
α	Thermal diffusivity, $\left(\frac{\mathrm{m}^2}{\mathrm{s}}\right)$ or $\left(\frac{\mathrm{ft}^2}{\mathrm{s}}\right)$
Δx	Spatial step, (s)
Δt	Temporal step, (s)
ΔT	Temperature change, (K) or (F)
$\frac{\partial T}{\partial t}$	Partial derivative of temperature with respect to t
$\frac{\partial^2 T}{\partial x^2}$	Second partial derivative of temperature with respect to x
ε	Material emissivity
σ	Boltzmann constant, $5.67051(10^{-8})\left(\frac{\mathrm{W}}{\mathrm{m}^2\,\mathrm{K}^4}\right)$ or $4.75644(10^{-13})\left(\frac{\mathrm{Btu}}{\mathrm{s\,m}^2\,\mathrm{K}^4}\right)$
ρ	Material density, $\left(\frac{\mathrm{kg}}{\mathrm{m}^3}\right)$ or $\left(\frac{\mathrm{lbm}}{\mathrm{ft}^3}\right)$

SUMMARY

A method for integrating Aeroheating analysis into conceptual reusable launch vehicle RLV design is presented in this thesis. This process allows for faster turn-around time to converge a RLV design through the advent of designing an optimized thermal protection system (TPS). It consists of the coupling and automation of four computer software packages: MINIVER[1], TPS*X*[2], TCAT[3] and ADS[4]. MINIVER is an Aeroheating code that produces centerline radiation equilibrium temperatures, convective heating rates, and heat loads over simplified vehicle geometries. These include flat plates and swept cylinders that model wings and leading edges, respectively. TPS*X* is a NASA Ames material properties database that is available on the World Wide Web. The newly developed Thermal Calculation Analysis Tool (TCAT) uses finite difference methods to carry out a transient in-depth 1-D conduction analysis over the center mold line of the vehicle. This is used along with the Automated Design Synthesis (ADS) code to correctly size the vehicle's thermal protection system (TPS). The numerical optimizer ADS uses algorithms that solve constrained and unconstrained design problems. The resulting outputs for this process are TPS material types, unit thicknesses, and acreage percentages.

TCAT was developed for several purposes. First, it provides a means to calculate the transient in-depth conduction seen by the surface of the TPS material that protects a vehicle during ascent and reentry. Along with the in-depth conduction, radiation from the surface of the material is calculated along with the temperatures at the backface and interior parts of the TPS material. Secondly, TCAT contributes added speed and automation to the overall design process. Another motivation in the development of TCAT is optimization. In some vehicles, the TPS accounts for a high percentage of the overall vehicle dry weight, as shown in Table 1. Optimizing the weight of the TPS will

thereby lower the percentage of the dry weight accounted for by the TPS. Also, this will lower the cost of the TPS and the overall cost of the vehicle.

Table 1. Dry Weight Percentage of TPS for Several Vehicles.

Vehicle	% of Dry Wt.	Dry Weight (lbs)
Hyperion[5]	6	123,250
Stargazer[6]	14	34,750
Shuttle[7]	16	154,739

In order to provide a proof of concept for the TCAT heating analysis tool, three test cases were considered. The first case was a reinforced carbon carbon (RCC) tile analyzed in a vacuum with constant convective heat rate to the top surface. The second and third test cases were both transient, trajectory-based analyses utilizing the STS-1 reentry flight profile.[8]

Along with the above case, benchmark solutions for the transient, trajectory-based cases were created using the computer heating analysis code SINDA[9]. The results of the benchmark solutions compared well with the results produced by TCAT for the second and third test cases.

In addition, a numerical analysis of the accuracy and execution time for the TCAT heating code was conducted by performing a sweeping analysis of the time step and spatial resolution. Results showed that the time required to perform the heating analysis at a single body point on a given geometry could be lowered from three minutes for an accurate solution to approximately 30 seconds with only a ten percent loss in accuracy.

In addition, an interface was created for the World Wide Web (WWW) that allows users the ability to size a thermal protection system for a reusable launch vehicle design from three different groups of TPS materials. The first group of materials are Shuttle technology materials with the second group consists of next generation RLV materials. The third group is composed of materials from the Shuttle materials group and

the next generation RLV material family. This interface was created using common gateway interface (CGI) scripts written in the Perl[10] programming language. These scripts allow for quick execution times that greatly reduce the amount of time required to size a thermal protection system.

Before scripting of the TPS sizing process was available, it required approximately eight man-hours to complete a TPS design for a representative conceptual RLV. Example cases examined using the WWW interface for TCAT were completed in a matter of minutes.

In addition, test scenarios for the WWW interface were developed and studied. The first test case involved TPS designs for a 10° half-angle cone, and the second case analyzed TPS designs on a multiple angle wedge with angles of 5° and 10°. The designs involved sizing thermal protection systems from each of the material groups available on the WWW interface.

Results of these test cases showed promising values for the average unit weights calculated. The tile materials that were chosen for the design test cases had unit weight values that ranged from 1.4~1.6 lbm/ft^2, and the blanket material TPS designs resulted in average unit weights between 0.4~0.6 lbm/ft^2.

CHAPTER I

INTRODUCTION

Thermal protection system (TPS) sizing calls for the selection of materials and a design that effectively protects the space vehicle and its cargo/passengers from the severe heating environment encountered during reentry and ascent. The overall design process involves several levels: conceptual, preliminary, and detailed design. At the conceptual level, ideas and assumptions are explored using engineering level tools that provide zeroth order knowledge of the overall design. Trade studies are conducted to test the feasibilities and sensitivities of the design space. In preliminary design, higher fidelity tools enable the designer to narrow down the domain of the design space. In detailed design, the design space is set, and a full accounting system is setup to accurately measure the effects of the assumptions made on the design.

In addition, the level of fidelity of the tools differs for each of the design levels. For example, in conceptual design a one-dimensional engineering heating code that takes several minutes to execute on a desktop PC is acceptable to size the thermal protection system for a reusable launch vehicle. On the other hand, preliminary and detailed design require experimental data for design verification and high fidelity computational fluid dynamic CFD codes that can solve the 3-D Navier-Stokes equations. These are both costly in terms of money and man-hours.

The work in this thesis is at the level of conceptual design, and explains a solution method that integrates Aeroheating analysis into conceptual RLV design. Chapter II will give an overview of recent and current TPS design strategies used in industry. Chapter III discusses the differences between "static, off-line" and "dynamic, on-line" TPS sizing. The "static, off-line" sizing process calls for making engineering assumptions in the early part of the design process that do not change, while "dynamic, on-line" sizing calls for

continuous updating of engineering assumptions. Chapter IV covers the computational tools required, MINIVER, TPS*X*, TCAT and ADS, in order to carry out the "dynamic, on line" TPS sizing strategy. Chapter V gives a detailed discussion of the theory utilized in the development of TCAT, and Chapter VI demonstrates several proof of concept studies and benchmarking results for TCAT. Chapter VII covers the issues of numerical accuracy and execution time of TPS designs using TCAT. Chapters VIII and IX focus on the development of a World Wide Web interface and design applications of the TCAT tool, respectively. Finally, conclusions and recommendations are given in Chapter X.

CHAPTER II

THERMAL PROTECTION SYSTEM DESIGN: AN OVERVIEW

The thermal protection system is one of the main focal points of any reusable launch vehicle design. This is true of the rockets from the early days of manned space flight to vehicles on the drawing board today. What is relevant are the methods used to design and verify the TPS for a RLV concept. With this in mind, this chapter will briefly cover a few of the methods that have been used and are currently in practice for TPS design.

One such method was used for the Space Shuttle TPS.[11] Orbiter flight test (OFT) data from the second flight of the Shuttle Orbiter (STS-2) was compared to the predictions of a three-dimensional thermal math model (TMM). This comparison resulted in the performance verification of the Shuttle's reusable surface insulation (RSI). The OFT data were obtained from RSI plug and gap thermocouple instrumentation. A schematic showing the RSI thermocouple instrumentation is shown in Figure 1. Quarter-tile RSI TMMs were developed using measured flight data for surface temperature and pressure environments. Extensive effort went into the development of models that accurately assessed the thermal performance of the RSI. Adequate thermal math model prediction techniques had to be created to compare with both ground test and flight data. Data gathered from STS-1 and STS-2 proved that mechanical and thermal requirements of the TPS were met and exceeded.[12] Other methods for the development of TMMs similar to this one were used to compare with arcjet ground test data.[13, 14]

A more recent approach for TPS design is a solution strategy developed to determine the thermal response of nonablative materials at hypersonic speeds formulated by Y.K. Chen and Frank S. Milos.[15] This procedure was developed for the design of thermal protection systems on reusable space vehicles and was motivated by the NASA

Access-To-Space Program.[16] Solutions were obtained through iterations between a finite element thermal response code (COSMOS/M)[17] and a finite volume reacting flow code (GASP).[18] This design procedure demonstrated multi-dimensional computations for both steady-state and transient trajectory-based conditions. Converged solutions were obtained in two to three global iterations using the properties of a radiative equilibrium wall as an initial guess for the process. The results found that temperature changes calculated between the first and last iterations were small. Therefore, it was concluded that the results after a single iteration were reasonably acceptable, and further iterations were not needed due to costly and time consuming multi-dimensional computations.

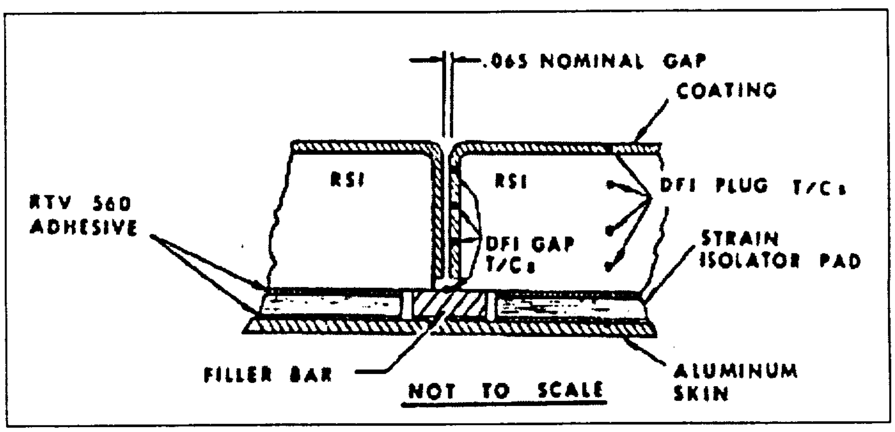

Figure 1. Schematic of RSI Thermocouple Instrumentation.[11]

The TPS design approach developed by Chen and Milos laid the foundation for the development of the TCAT design tool. As mentioned, the thermal response of the materials was obtained by the loosely coupled iterations between COSMOS and GASP, which are both high fidelity, powerful, and computationally expensive flow codes used in preliminary and advanced design. The TPS design process discussed in this thesis utilizes a strategy similar to that of Chen and Milos, but the computational tools utilized are at the engineering level and are more suitable for conceptual level design.

Another method of TPS design is that used for the X-33.[19] This method uses a trajectory-based thermal protection system sizing strategy. The general purpose of the X-33 program is to build a subscale, sub-orbital, reusable, single-stage-to-orbit (SSTO), technology demonstrator RLV. The TPS sizing strategy for the X-33 involves the discretization of the heat rate at eight points along the sub-orbital trajectory. Computational fluid dynamic calculations are used to create an aerothermal database for laminar and turbulent flow conditions. Several TPS sizings were generated using this database and thermal protection system stackups. Examples of the stackups are shown in Figure 2.

From this work, it was concluded that a baseline trajectory, including both laminar and turbulent flow calculations, produced a TPS material map that was within the thermal envelopes of all materials for nearly the entire vehicle surface.

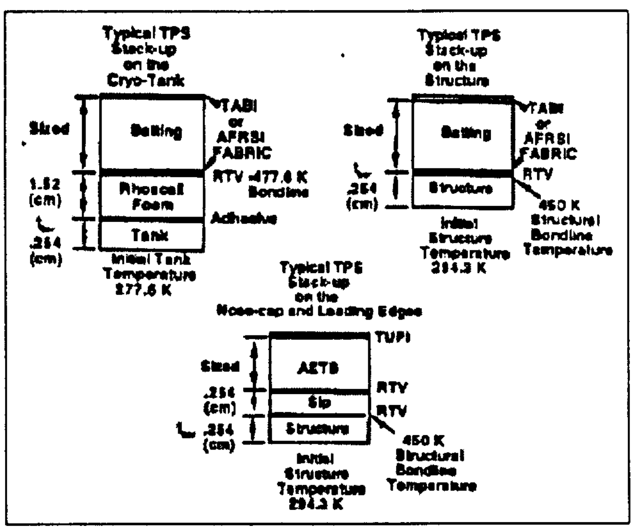

Figure 2. One-Dimensional TPS Stackups.[19]

The procedures that have been discussed in this section involve great detail and require high fidelity calculations in order to achieve the desired accuracy. It is important to point out that this level of design is called "detailed" or "preliminary" design. For this

thesis, the focus will be at the conceptual level. At the phase, design tools used typically run in the order of minutes. This allows to a quick turn-around time for vehicle designs and enables the designer to carry out trade studies.

CHAPTER III

STATIC VERSUS DYNAMIC TPS SIZING

This chapter will discuss two different TPS design strategies: static and dynamic TPS sizing. Static TPS sizing calls for assumptions to be made prior to the design iteration process. These assumptions remain constant during the design of a vehicle. An example is a constant unit weight. Dynamic TPS sizing allows these assumptions made a priori in the design process to be updated throughout the iteration cycle. Dynamic TPS sizing is preferred because it provides the ability to update the assumptions during the iteration process, allowing for better confidence in the final solution.

Static Off-Line TPS Sizing

In many design organizations, Aeroheating analysis is done in the conceptual design phase. In some cases, it is an off-line static process where single point assumptions are made about TPS unit weights, thicknesses, and acre percentages. These single point assumptions are set at the beginning of the design and are not changed throughout the convergence process. Consequently, there is no communication between the Aeroheating and Aerodynamics disciplines as shown in the Design Structure Matrix (DSM) of Figure 3.

The DSM is a way to show discipline interaction in a design environment. It shows a cascading arrangement of the design disciplines in the form of a diagonal. Feed forward links above the diagonal represent the progression of information, inputs, from one discipline to next. Below the diagonal are feed back links. These are used to pass information from a discipline lower in the DSM to one that is in a higher position. Feedback is important because it provides the flexibility of changing the design when required. In addition, disciplines are related through strong and weak links. The strong

links are indicated by the black dots and the weak links are designated by the gray dots. Strong links imply that the communication between two disciplines is important in the outcome of the design or that a local iteration is required before information can be passed to the next discipline. Weak links require communication, but they are not as important as the strong links.

The feedback from Aeroheating to Aerodynamics is important; for example, the leading edges of the vehicle may be too small, creating a heat load that is too high, based on the assumed TPS. To lower the heat load, the Aeroheating analyst desires blunt leading edges. Aerodynamicists desire sharp, small radii leading edges that minimize drag. The objective of these two disciplines conflict with each other; therefore, there is an optimum leading edge value that must be found so that both disciplines are satisfied. Integrated Aeroheating will help find this optimum value.

Another consequence that arises when making these off-line assumptions is that there is no interaction between the Aeroheating and Weights and Sizing disciplines. This is similar to the situation between Aeroheating and Aerodynamics where an optimum value must be reached. Here the optimum is to design a TPS structure with the minimum unit weight. If fixed assumptions are made at the beginning of the design phase, then optimum values cannot be obtained. Also, conservative assumptions may produce a TPS that is too thick and heavy. On the other hand, on the other hand, nonconservative assumptions may produce a TPS that will ineffectively protect the RLV from aeroheating effects induced by the flow field.

Static off-line TPS sizing is a one-and-done endeavor. Several tools have to be manually run before finalized values of the thicknesses, unit weights, and acre percentages are determined. The first tool is MINIVER, which is an Aeroheating analysis code written by NASA. It produces centerline radiation equilibrium temperature distributions, convective heating rates, and heat loads over simplified vehicle geometries. These geometries include flat plates to model wings and swept cylinders to model leading edges. The constituent materials of the TPS are determined by the radiation equilibrium temperatures produced by MINIVER and are selected from the NASA Ames TPS*X*

material database. It is important to make sure the multiuse temperature limit of the selected material is greater than the radiation equilibrium temperature obtained from MINIVER. Also, it is desired that the chosen material is lightweight and has low maintenance along with minimal inspection requirements.

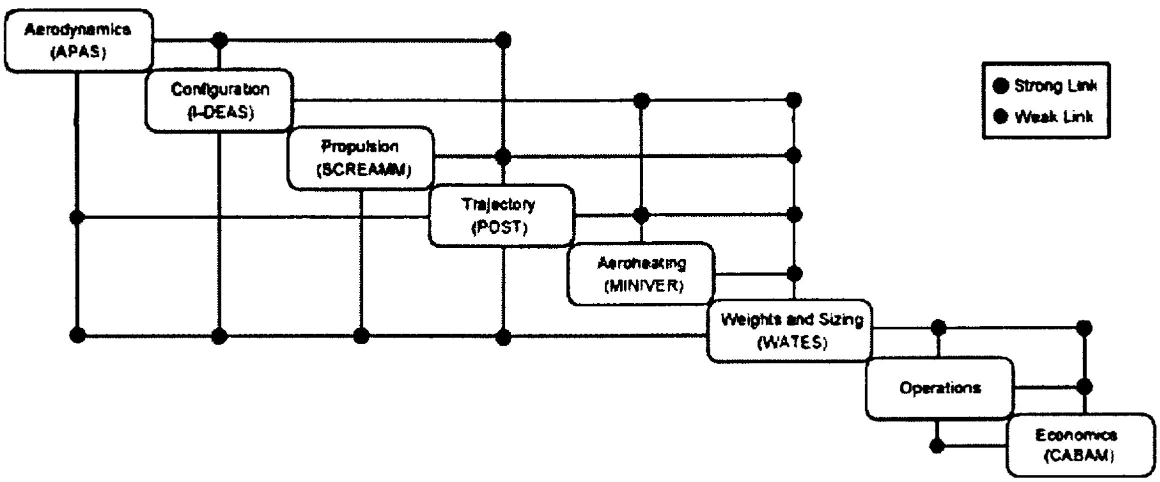

Figure 3. Static TPS Strategy DSM.

After running MINIVER and obtaining the appropriate material properties list from TPS*X*, the TPS unit weights, thicknesses, and acre percentages are estimated. The actual values for the unit weights can be estimated by basing them on vehicles of a similar design. A deficiency with this approach is that the reference vehicle will most likely have a different trajectory. This could lead to two possible problems. First, the heat load applied to the vehicle for the current design might be too high. This means the unit weight values obtained from the reference design are underestimated. Secondly, the opposite can be true where the unit weights are overestimated and cause the overall weight of the RLV to be higher than necessary.

This process of TPS sizing with MINIVER and TPS*X* is conducted at several different points of interest on the vehicle, which can significantly increase the total time

required to design a TPS. Without iteration, the TPS unit weights and acreage areas become fixed for the rest of the design process.

There are several reasons why this process needs to be improved. First, static off-line TPS sizing may involve making faulty assumptions that will lead to non-optimal design results. Secondly, there is no coupling mechanism that allows an analysis of the in-depth conduction based on the convective heating rates obtained from MINIVER. A conduction analysis would allow the capability of changing the TPS design. Thirdly, the process is too time-consuming; it might take several days, which would delay the turn-around time for a vehicle design. All these lessons call for the coupling of TPS sizing into a design oriented structure. These changes lead to dynamic on-line TPS sizing.

Dynamic On-Line TPS Sizing

Figure 4 shows how the DSM changes in the Dynamic TPS sizing strategy compared to that of the static TPS design strategy. Now there are feedback links to Aerodynamics, Trajectory, and Configuration from the Aeroheating discipline. Also, there is a feedback link from Weights and Sizing to Aeroheating. These provide communication from Aeroheating that was not available in the static TPS design strategy.

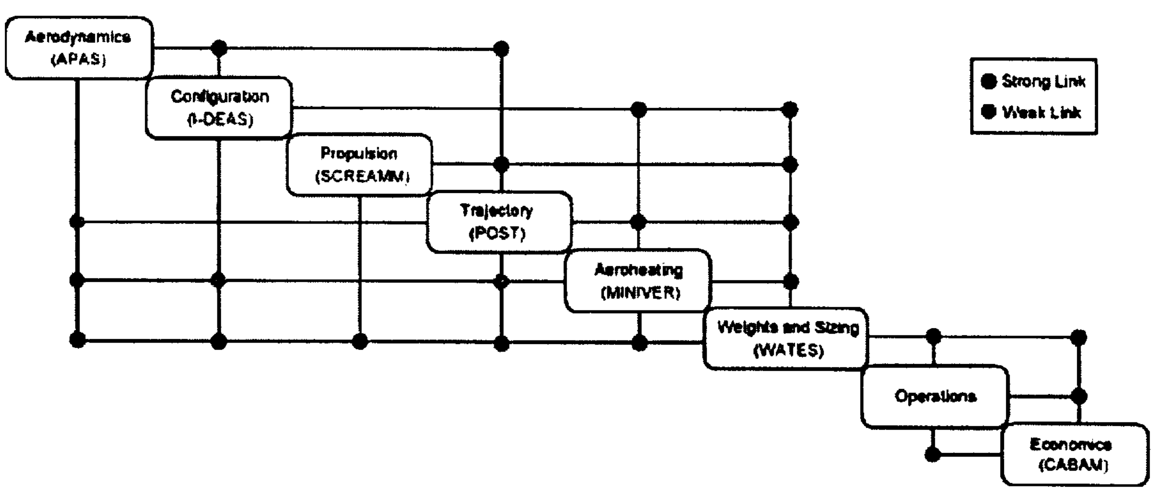

Figure 4. Dynamic TPS Sizing Design Structure Matrix.

The integration of the feedbacks into the DSM enables TPS sizing to be a dynamic process. This means the sized TPS values are revised and optimized during each iteration until the vehicle is converged. This integrated approach will involve the coupling of four tools: MINIVER, TPS*X*, TCAT and ADS.

Descriptions of MINIVER and TPS*X* items were provided earlier. TCAT, the Thermal Calculation Analysis Tool, is an original code written for this research that uses finite difference methods coupled with optimization techniques in order to conduct a transient, trajectory-based heating analysis to design and size the TPS of an RLV. The Automated Design Synthesis (ADS) tool uses algorithms that can solve constrained and unconstrained design optimization problems.

The diagram in Figure 5 gives a pictorial representation of the dynamic on-line TPS sizing process. First, a trajectory file from POST[20] (Program To Simulate Optimized Trajectories) is obtained. This raw trajectory file includes time, altitude, velocity, angle-of-attack, and sideslip angle in column format. It is necessary to thin the trajectory, remove points from the file, if it contains more than fifty points. The program *thindata* is used to reduce the number of points to 50 or less. The removal of points is based on the gradient of the curve in question. More points are left where the slope is rapidly changing and points are removed in areas where the gradient is relatively constant. *Thindata* creates an output file "tmp.john" that is used to transfer the trajectory information from POST to MINIVER. Once the MINIVER deck is created, it is analyzed using *lanmin91*. A 2-D Aeroheating analysis is conducted and produces convective heating rates, heat loads, and radiation equilibrium temperatures. These are calculated using empirical methods such as the Fay-Riddell stagnation point method and the Eckert's reference enthalpy method for flat plate heating. Then, a convective heat rate vs. time array obtained from the MINIVER output and a material properties list is input into TCAT. In turn, TCAT and ADS numerically optimize the thickness, unit weight, and acreage percentage values of the TPS at each selected body point of the vehicle. Specifically, the optimizer minimizes the unit weights (thickness) based on several constraints, and multi-use temperature limits. The constraints include maximum surface temperature, maximum

backface temperature, and maximum material interface temperature limits. Once the point calculations are completed, aggregate values for the acre percentages for each TPS material used are calculated for the vehicle. These properties are then passed on to the Weights and Sizing discipline.

This process would normally be time consuming, but through scripting and automation of the data transfer, the manual work required by the user is significantly reduced. Therefore, there are several advantages to having a dynamic on-line TPS sizing process. First, it clearly provides a means of reaching near optimal values for the vehicle's TPS at each design iteration. In addition, it allows for a quicker turn-around time for a TPS design as compared to the static off-line process. TCAT is a tool that makes dynamic TPS sizing possible.

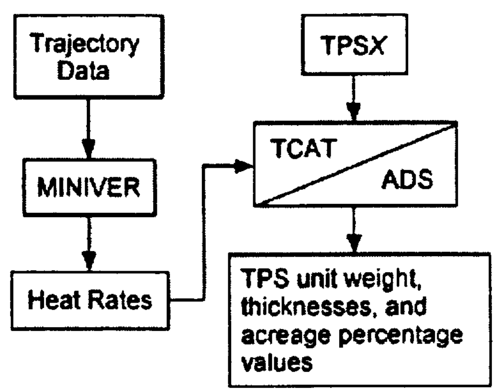

Figure 5. Dynamic On-Line TPS Sizing Process.

CHAPTER IV

COMPUTATIONAL TOOLS USED IN DYNAMIC TPS SIZING

This section will discuss the tools used in the solution strategy that enables thermal protection system sizing to be integrated into conceptual level design. The tools discussed are MINIVER, TPS*X*, ADS, and TCAT.

MINIVER

MINIVER is the miniature version of the JA70 General Aerodynamic Heating Computer Code. It is a FORTRAN legacy code that was developed in the early 1970s to provide an engineering level Aeroheating analysis capability.

At its roots, MINIVER is an approximate engineering code used to assess the Aerothermal environment over critical regions of a reusable launch vehicle. The required inputs for MINIVER are altitude, velocity, angle-of-attack, and sideslip angle as functions of time obtained from the trajectory analysis along with the geometric configuration of the vehicle. Using these inputs, MINIVER models the local flowfield conditions using shock angles and pressures based on wedge, cone, or parallel shock approximations such as Newtonian pressures. MINIVER applies standard engineering techniques to calculate heating over critical areas of the vehicle including:

- Nose cap stagnation area – Fay-Riddell Stagnation point heating equation
- Windward centerline – Eckert's Reference Enthalpy Flat Plate Method
- Wing and fin leading edges – Modified Beckwith/Gallagher Swept Cylinder Method
- Deflected control surfaces – Bushnell and Weinstein, Flap Reattachment Heating
- Leeward surface area – Bertin and Goodrich, Leeside Orbiter Heating

Use of these models and applications results in critical Aeroheating parameters including laminar and turbulent heat transfer rates, heat loads, radiation equilibrium temperature, local and wall conditions, and transition information.

TPS*X*

TPS*X* – Thermal Protection System Expert is a material database created by the Thermal Protection Materials and Systems Branch at NASA Ames Research Center. It exists to maintain properties of advanced thermal protection system (TPS) materials, and it is available on the Internet. It provides an easy user interface for retrieving, displaying, and graphing information in a variety of formats. The database includes thermal, mechanical, ablation, and chemical properties, along with cost data. It also provides description of each material, including references and common applications. TPS*X* also contains photos or sketches of some materials. Materials in the material database include:

- Rigid Tiles
- Flexible Blankets
- Carbon-Based Reusable Ceramic Composites
- Ultra-High Temperature Composites
- Adhesives
- Structural Organic Composites

ADS

ADS - The Automated Design Synthesis tool is a FORTRAN code used for the solution of linear and nonlinear, unconstrained and constrained optimization problems. ADS is segmented into three different levels: strategy, optimizer, and one-dimensional search. Sequential unconstrained minimization, the augmented Lagrange multiplier method, and sequential quadratic programming are three examples of methods available at the strategy level. Two examples at the optimizer level are variable metric methods, and the method of feasible directions. Lastly, the Golden Section method and polynomial interpolation are examples available at the one-dimensional search level. Different

combinations of the methods on each level allow for many different types of problem solutions.

TCAT

TCAT - The Thermal Calculation Analysis Tool is a FORTRAN code created specifically as the centerpiece tool to facilitate the integration of Aeroheating analysis into conceptual design framework for RLVs. It provides a means to calculate the transient in-depth conduction at the surface of a TPS material that protects a vehicle during ascent and reentry. Along with the in-depth conduction, radiation from the surface of the material is calculated along with the temperature at the backface of the TPS material. The next chapter provides a detailed discussion of the theory and assumptions used in the development of TCAT.

CHAPTER V

THE THERMAL CALCULATION ANALYSIS TOOL (TCAT)

The previous chapter described the roles of MINIVER, TPS*X*, and ADS in the process of integrating Aeroheating analysis into conceptual RLV design. Also, a brief description of TCAT was given, but further discussion is necessary. Therefore, a stand-alone chapter for the discussion of the theory used in the development of TCAT is needed for two reasons. First, TCAT is the primary computational tool used in the dynamic on-line TPS design process. Secondly, it is an original code created to provide an in-depth, transient conduction analysis that drives the design of the RLV TPS. This chapter will discuss the theory and assumptions used in order to conduct the heating analysis provided by TCAT.

TCAT uses a fully implicit method in order to solve the parabolic, one-dimensional unsteady heat conduction equation (1) by marching in time.

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \tag{1}$$

The boundary conditions given in equations (2) and (3) are applied to the top and bottom surfaces of the TPS material.

$$q_{conv} - \varepsilon \sigma T_s^4 + k \frac{dT}{dx} = 0 \quad \text{at} \quad x = 0 \tag{2}$$

$$\frac{dT}{dx} = 0 \quad \text{at} \quad x = L \tag{3}$$

The top surface is defined as $x = 0$, and $x = L$ is at the backface. Equation (2) is the energy balance relationship for the top surface of the TPS material; it includes convection from the flow field, radiation from the heated surface, and conduction absorbed by the TPS material. All these quantities are summed to equal zero in order to preserve the conservation of energy. Equation (3) states that there is an adiabatic wall at the backface of the material. This is a conservative assumption that is used in order to model an insulator. This assumption resembles an insulator by allowing the temperature at the backface of the material to increase or decrease over time due to the absorption or release of energy from the material. This assumption is conservative because the temperature rise and decay of an adiabatic wall does not fully model the heat capacitance of the actual structure that physically exists behind the TPS material. This type of modeling would require methods of analysis that are not within the scope of this thesis. Furthermore, modeling the thermal response of the vehicle structure is beyond the scope of conceptual design.

Three different types of discretization were used for obtaining the system of equations needed to solve for the in-depth conduction analysis. A forward implicit difference scheme was used at the top surface in order to incorporate the boundary condition given in equation (2). The mesh map for the top surface can be seen in Figure 6. At the top surface node, the second derivative term in the heat equation was obtained through a Taylor series expansion and incorporation of the top surface boundary condition. This discretization resulted in equation (4) and is accurate on the order of $O(\Delta t, \Delta x)$.

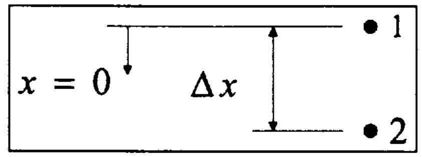

Figure 6. Nodal Mesh Map for Top Surface Node.

$$T_1^n = \frac{2\alpha_1 \Delta t}{\Delta x}\left(\frac{\varepsilon_1 \sigma \left(T_1^{n+1}\right)^4 - q_{conv}}{k_1}\right) - \frac{2\alpha_1 \Delta t}{\Delta x^2}\left(T_2^{n+1} - T_1^{n+1}\right) + T_1^{n+1} \tag{4}$$

The heat equation was discretized at the interior nodes, using the nodal map shown in Figure 7, with a simple implicit central finite difference scheme resulting in equation (5) that is accurate on the order of $O(\Delta t, \Delta x^2)$.

$$T_i^n = -\frac{\alpha_i \Delta t}{\Delta x^2}\left(T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1}\right) + T_i^{n+1} \tag{5}$$

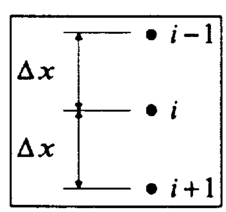

Figure 7. Nodal Mesh Map for Interior Nodes.

On the back surface of the material, an implicit backward finite difference scheme was used to discretize the heat equation and couple the boundary condition from equation (3). This resulted in equation (6), which is accurate on the order of $O(\Delta t, \Delta x)$. Figure 8 shows the nodal mesh.

$$T_N^n = -\frac{2\alpha_N \Delta t}{\Delta x^2}\left(T_{N-1}^{n+1} - T_N^{n+1}\right) + T_N^{n+1} \tag{6}$$

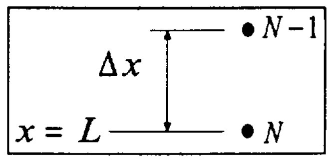

Figure 8. Nodal Mesh Map for Backface Node.

A radiative heat flux at the backface of the material can be used as an alternative boundary condition instead of the adiabatic wall assumption. This is given by equation (7), which states that the radiative heat flux at the backface is equal to the conductive heat flux trough the material. The conductive heat flux term is given by Fourier's Law and the radiative flux is a function of the backface surface emissivity, Boltzmann's constant, and the backface surface temperature. The result of the discretized heat equation at the backface incorporating this alternative boundary condition is given by equation (8) and accurate on the order of $O(\Delta t, \Delta x)$. This alternative boundary condition is presented because it used later in Chapter VI in a proof of concept case study for the TCAT tool.

$$-k_N \frac{dT}{dx} = \varepsilon_N \sigma T_N^4 \tag{7}$$

$$T_N^n = -\frac{2\alpha_N \Delta t}{\Delta x^2}\left(T_{N-1}^{n+1} - T_N^{n+1} - \frac{\varepsilon_N \sigma}{k \Delta x}\left(T_N^{n+1}\right)^4\right) + T_N^{n+1} \tag{8}$$

A system of nonlinear equations (9) resulted once the heat equation was discretized at all nodes in the material. This system of equations reflects the use of the top surface and backface boundary conditions given by equations (2) and (3) respectively. This system is nonlinear due to the fourth order radiation term in the top surface boundary condition, equation (2). Each equation in this system represents a nodal location in the material. Each is obtained by subtracting the information at the current time level, n, from that required at the next time level, $n+1$, and setting it equal to f_i, where $i = 1,..,N$, with N being the maximum number of nodes.

$$
\begin{aligned}
f_1 &= \left(1+\frac{2\alpha_1\Delta t}{\Delta x^2}+\frac{2\alpha_1\Delta t}{k_1\Delta x}\varepsilon\sigma\left(T_1^{n+1}\right)^3\right)T_1^{n+1}-\frac{2\alpha_1\Delta t}{\Delta x^2}T_2^{n+1}-\left(T_1^n-\frac{2\alpha_1\Delta t}{\Delta x k_1}q_{conv}\right)\\
f_2 &= -\frac{\alpha_2\Delta t}{\Delta x^2}T_1^{n+1}+\left(1+\frac{2\alpha_2\Delta t}{\Delta x^2}\right)T_2^{n+1}-\frac{\alpha_2\Delta t}{\Delta x^2}T_3^{n+1}-T_2^n\\
f_3 &= -\frac{\alpha_3\Delta t}{\Delta x^2}T_2^{n+1}+\left(1+\frac{2\alpha_3\Delta t}{\Delta x^2}\right)T_3^{n+1}-\frac{\alpha_3\Delta t}{\Delta x^2}T_4^{n+1}-T_3^n\\
&\quad\vdots \qquad\qquad\quad \vdots \qquad\qquad\quad \vdots\\
f_{N-1} &= -\frac{\alpha_{N-1}\Delta t}{\Delta x^2}T_{N-2}^{n+1}+\left(1+\frac{2\alpha_{N-1}\Delta t}{\Delta x^2}\right)T_{N-1}^{n+1}-\frac{\alpha_{N-1}\Delta t}{\Delta x^2}T_N^{n+1}-T_{N-1}^n\\
f_N &= -\frac{2\alpha_N\Delta t}{\Delta x^2}T_{N-1}^{n+1}+\left(1+\frac{2\alpha_N\Delta t}{\Delta x^2}\right)T_N^{n+1}-T_N^n
\end{aligned}
\tag{9}
$$

This system of equations is iteratively solved using the Newton-Raphson method, which is the application of Newton's root solving method applied to a system of non-linear equations. The first step in the Newton-Raphson method requires the formation of the Jacobian matrix, J, which is defined by equation (10).

$$J_{i,j} = \frac{\partial f_i}{\partial T_j^{n+1}}_{\; i,j=1,\dots,N} \tag{10}$$

Once the Jacobian is formed, the problem takes on the form $Ax = b$ given by equation (11), where A is the Jacobian and is a tridiagonal matrix, x is the change in temperature at time level $n+1$, and $b = -f$ at time level n.

$$\begin{pmatrix} \frac{\partial f_1}{\partial T_1^{n+1}} & \frac{\partial f_1}{\partial T_2^{n+1}} & 0 & 0 & 0 & 0 \\ \frac{\partial f_2}{\partial T_1^{n+1}} & \frac{\partial f_2}{\partial T_2^{n+1}} & \frac{\partial f_2}{\partial T_3^{n+1}} & 0 & 0 & 0 \\ 0 & \frac{\partial f_3}{\partial T_2^{n+1}} & \frac{\partial f_3}{\partial T_3^{n+1}} & \frac{\partial f_3}{\partial T_4^{n+1}} & 0 & 0 \\ & & \ddots & \ddots & \ddots & \\ 0 & 0 & 0 & \frac{\partial f_{N-1}}{\partial T_{N-2}^{n+1}} & \frac{\partial f_{N-1}}{\partial T_{N-1}^{n+1}} & \frac{\partial f_{N-1}}{\partial T_N^{n+1}} \\ 0 & 0 & 0 & 0 & \frac{\partial f_N}{\partial T_{N-1}^{n+1}} & \frac{\partial f_N}{\partial T_N^{n+1}} \end{pmatrix} \begin{pmatrix} \Delta T_1 \\ \Delta T_2 \\ \Delta T_3 \\ \vdots \\ \Delta T_{N-1} \\ \Delta T_N \end{pmatrix}^{n+1} = \begin{pmatrix} -f_1 \\ -f_2 \\ -f_3 \\ \vdots \\ -f_{N-1} \\ -f_N \end{pmatrix}^{n} \tag{11}$$

This is solved by making an initial guess for the temperature at time level $n+1$, and iteratively solving for ΔT^{n+1} using the Thomas Algorithm[21]. Prior to the next iteration step the temperature at time level $n+1$ is updated by $T^{n+1} = T^{n+1} + \Delta T^{n+1}$.

Convergence is reached when the magnitude of the two-norm for both the $\boldsymbol{\Delta T^{n+1}}$ vector and the $\boldsymbol{f^n}$ vector on the right hand side of equation (11) fall below $1(10^{-6})$. The definition of the vector two-norm is given by equation (12). Also, the maximum number of iterations is limited to 1000.

$$\|g\|_2 = \left(\sum_{i=1}^{N} g_i^2 \right)^{1/2} \tag{12}$$

The initial guess for the temperature profile with in the material for the first $n+1$ time step is assumed to 1000 K. For each additional $n+1$ time step, the initial guess for the temperature profile is assumed to the final converged temperature profile from the previous time step.

TCAT can analyze up to 100 nodes in a single material or 100 nodes total when several disparate TPS materials are layered together. When different materials are layered together, it is assumed that perfect contact exists and equation (13) gives the interface condition for the heat transfer between the materials.

$$k_{i-1}\left(\frac{dT}{dx}\right)_{i-1} = k_i\left(\frac{dT}{dx}\right)_i \tag{13}$$

Also, it is assumed that no kinetic reactions occur in the boundary layer; therefore, chemical equilibrium exists, while thermal equilibrium does not. Additionally, all material properties are held constant throughout the analysis. At this time, temperature dependent material properties are not incorporated in TCAT, but can be added as linear interpolations or cubic spline functions.

CHAPTER VI

TCAT PROOF OF CONCEPT RESULTS

Case 1: Tile In a Vacuum

This case study was conducted in order to demonstrate that TCAT's heating analysis could model steady-state heat transfer. This case analyzed a 0.1016 m (4 in) thick reinforced carbon carbon (RCC) tile with an applied 200 kW/m^2 convective heat rate to the top surface of the material. The top surface boundary condition of the tile the same given by equation (2) in Chapter V. Recall, equation (2) is a surface energy balance that includes convection from the flowfield, radiation from the surface, and conduction through the material. For this case, the adiabatic wall assumption given by equation (3) was replaced by the conduction and radiative boundary condition given by equation (7). The material properties for RCC at 300 K are given in Table 2.

Table 2. Case 1 Assumptions.

Material	RCC
Density, ρ	1580 kg/m^3
Specific Heat, c_p	0.77 kJ/kg-K
Thermal Conductivity, k	4.3 W/m-K
Emissivity, ε	0.79

Figure 9 shows the temperature history profile for the RCC tile. Both the surface and backface temperatures attain steady state at approximately 1400 K and 800 K,

respectively. These temperatures are under the 1900 K multiuse temperature limit of RCC.

Figure 10 shows that all modes of heat transfer attain steady state. As expected, the surface conduction started equal to the value of the constant convection term and dropped off as the radiation from the surface increased. In addition, it can be seen that the radiative heat flux from the backface lags and is small relative to that on the top surface. This is due to the fact that most of the convective heat flux is reradiated away from the top surface. In comparison, the radiative heat flux from the top surface is approximately 175 kW/m^2, and the backface radiative flux reaches approximately 20 kW/m^2. Also, it can be seen that the conductive heat flux reaches an approximate steady-state value of 25 kW/m^2.

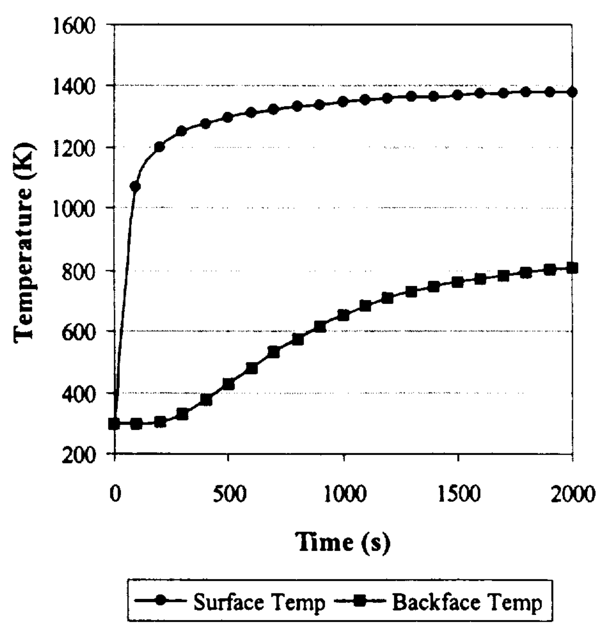

Figure 9. Temperature History Plot Case 1.

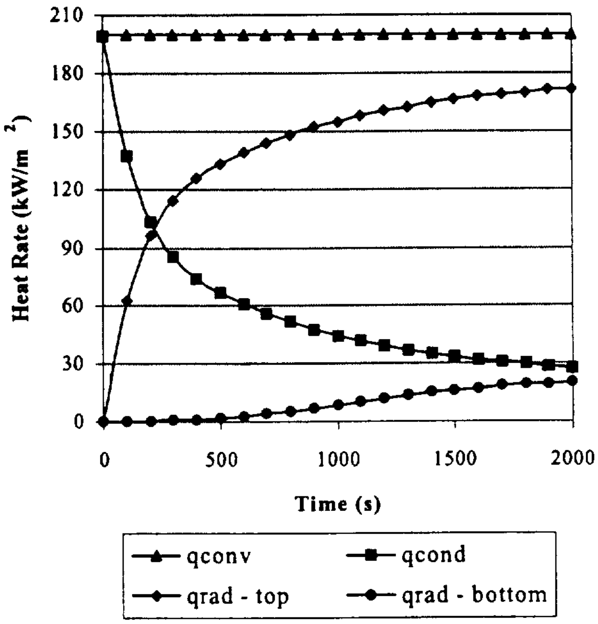

Figure 10. Heat Rate History Plot Case 1.

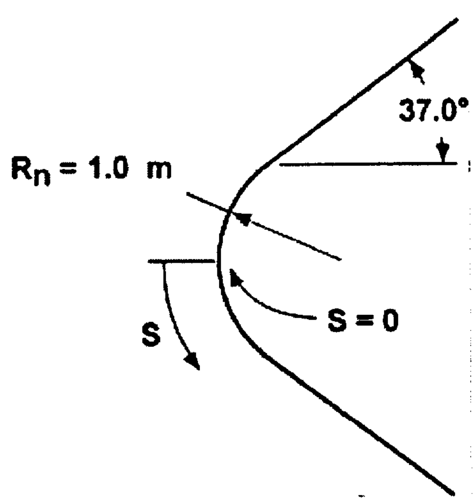

Figure 11. One-Meter Nose Radius 37° Half-Angle Spherical Cone.

Case 2: Trajectory-Based Transient Analysis

This analysis was conducted in order to provide a proof of concept for the trajectory-based transient analysis capability of TCAT. The blunted spherical-cone in Figure 11 was covered with 0.1015 m (4 in) thick RCC tiles and flown along the STS-1 descent trajectory. Figures 12–15 show pertinent information related to the STS-1 reentry profile. Altitude vs. time is shown in Figure 12; velocity vs. time is shown in Figure 13; angle of attack vs. time is shown in Figure 14, and Figure 15 shows altitude vs. velocity. Table 3 lists the material properties of RCC at 300 K.

Table 3. Case 2 Assumptions.

Material	RCC
Density, ρ	1580 kg/m^3
Specific Heat, c_p	0.77 kJ/kg-K
Thermal Conductivity, k	4.3 W/m-K
Emissivity, ε	0.79

Figures 16-21 show temperature and heat rate histories for three locations on the body of the blunted spherical-cone. The first is at the tip of the cone, S = 0; the second lies on the windward side, S = 0.925 m; and the third point is on the leeward side of the cone, S = – 0.925 m. "S" corresponds to running length along the cone starting from the tip point defined as S = 0.

In Figure 16, it can be seen that the maximum temperatures in the nose region, at S = 0, were approximately 1600 K and 700 K on the surface and backface, respectively. According to the TPS*X* database, these temperatures are clearly below the 1900 K multiple use limit of RCC. It is also clear that the surface temperature peaked before reaching 1900 K on the back surface. This demonstrates RCC's ability to act as an insulator. This is evident because the top surface temperature peaks and falls before that of the backface of the material. Also, the backface of the material does not see temperatures as high as those on the top surface.

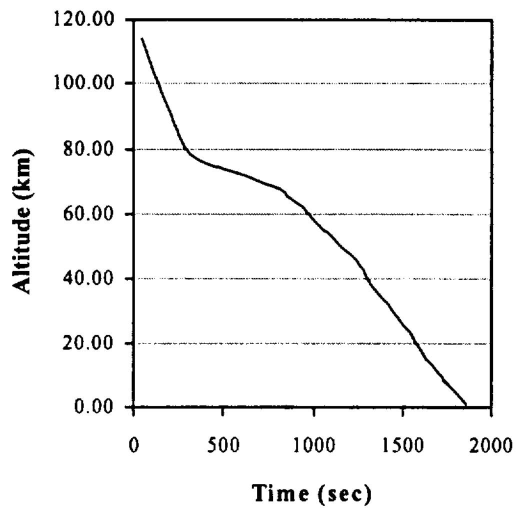

Figure 12. STS-1 Reentry Trajectory Altitude vs. Time.

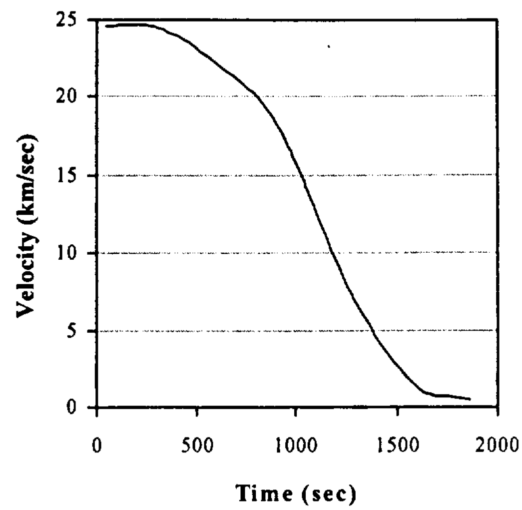

Figure 13. STS-1 Reentry Trajectory Velocity vs. Time.

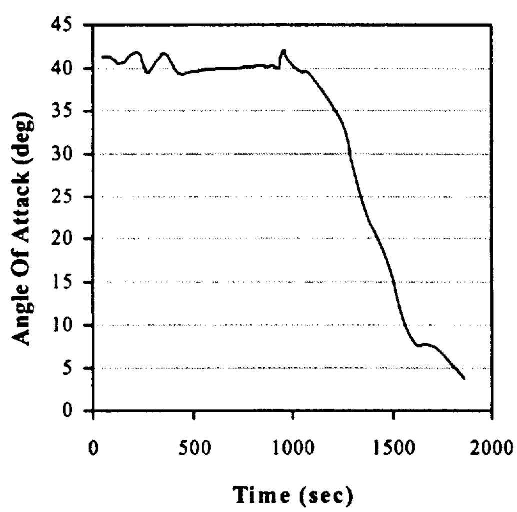

Figure 14. STS-1 Reentry Trajectory Angle Of Attack vs. Time.

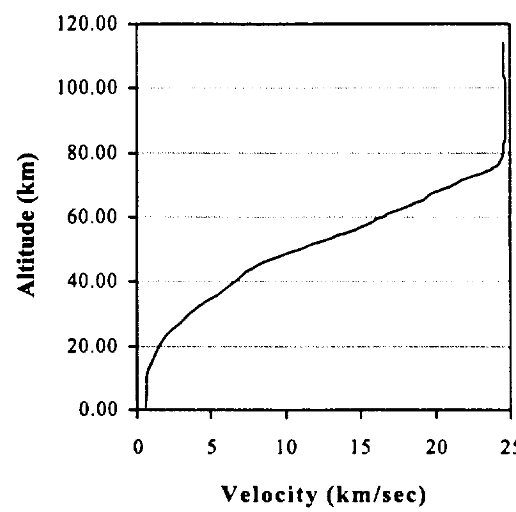

Figure 15. STS-1 Reentry Trajectory Altitude vs. Velocity.

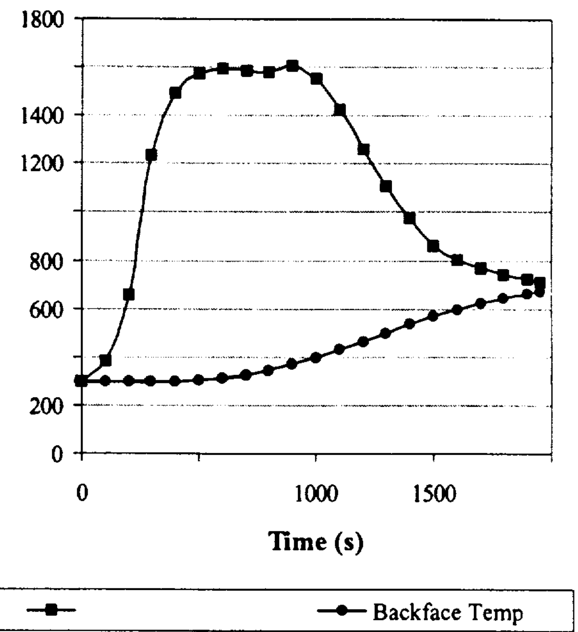

Figure 16. S = 0 Point Temperature History Case 2.

If the thickness of the RCC tile is increased, then it will take longer for the effects of the conduction into the material to reach the backface. This means the temperature value will be lower, and it will take longer before it reaches its peak value. The opposite would occur if the thickness of the RCC were decreased.

The surface heat rate histories for the nose region, S = 0, of the 37° half-angle cone are shown in Figure 17. The conductive heating rate follows the convective heating rate until the surface becomes hot enough to effectively radiate heat off the surface. This causes a slope change in the conductive heating rate. After 200 seconds, the conductive heating rate becomes negative at the point where the radiative heat rate away from the surface of the RCC becomes greater than that of the convective heating rate. This means that the flow field has become cooler than the surface of the material and heat is therefore being removed from the RCC material.

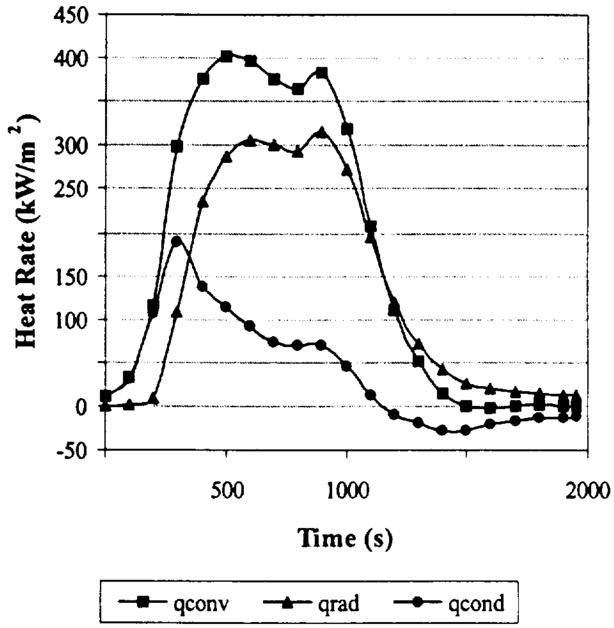

Figure 17. S = 0 Point Heating History Case 2.

Figures 18 and 19 show the temperature and surface heating histories for the point located on the windward side of the vehicle at a running length of S = 0.925 m. The maximum temperature attained at this point is approximately 1300 K, which is less than the maximum temperature for the nose region. Also, the surface heat rate values for all modes of heat transfer were less than those in the S = 0 region. This means this location receives less of a heat load, which leads to lower backface temperatures over the course of the trajectory. This occurs because the integrated heat load at the windward side location is less than that at the S = 0 point and the TPS thickness at both locations is the same. Therefore, the thickness of the RCC tile of the windward side location could be decreased because the TPS is overdesigned. If this were an actual vehicle design, the overdesign of the TPS would result in unnecessary TPS weight and higher overall cost associated with the vehicle.

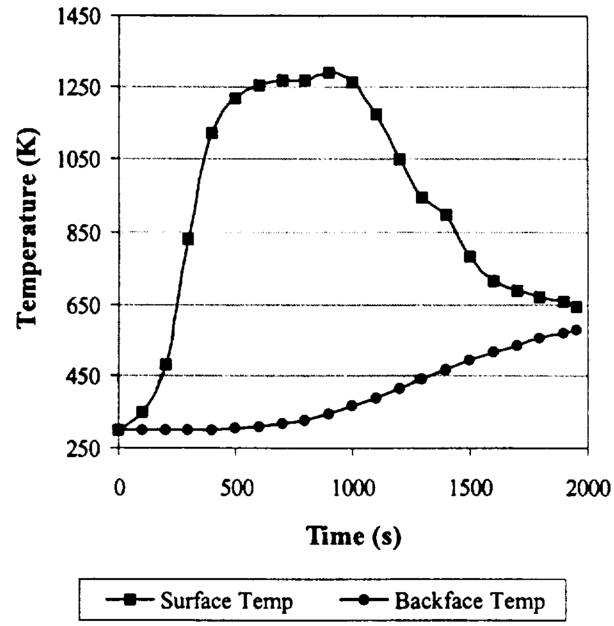

Figure 18. Windward Body Point Temperature History Case 2.

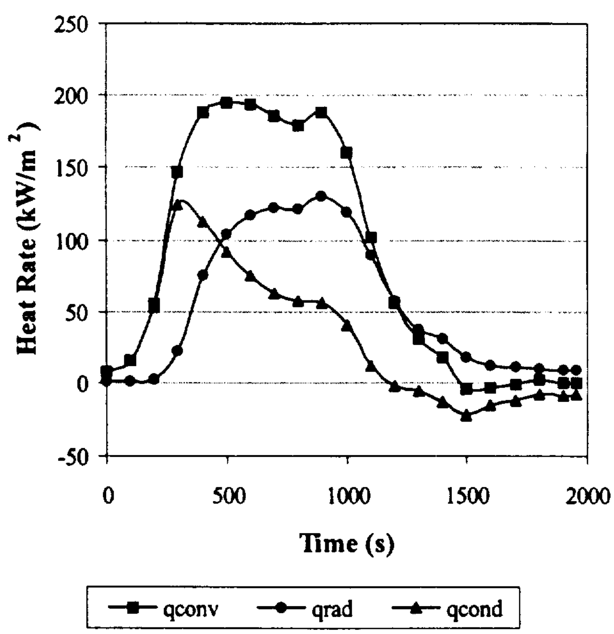

Figure 19. Windward Body Point Heating History for Case 2.

Figures 20 and 21 show the temperature and surface heating rate histories at a point on the leeward side of the blunted spherical-cone. It is evident that the same trends are found here as those for the S = 0 region and windward side body locations discussed earlier. Also, the temperature and surface heat rate values are lower than the other two body points because the point is located on the leeward side of the vehicle mold line. On the leeward side, shocks are weaker when positive angles of attack are sustained, as is the case with the STS-1 descent trajectory of Figure 18.

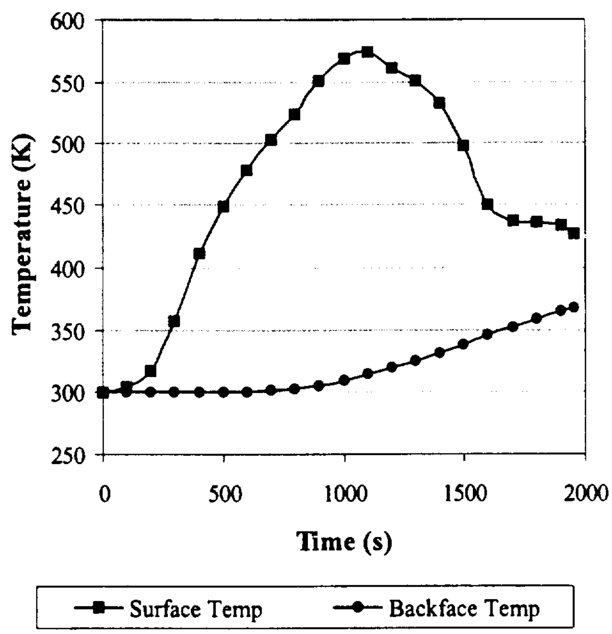

Figure 20. Leeward Body Point Temperature History Case 2.

Figure 22 gives a comparison of the temperature histories at each of the three points considered, and reinforces the fact that the S = 0 region has greater temperature values than the others considered.

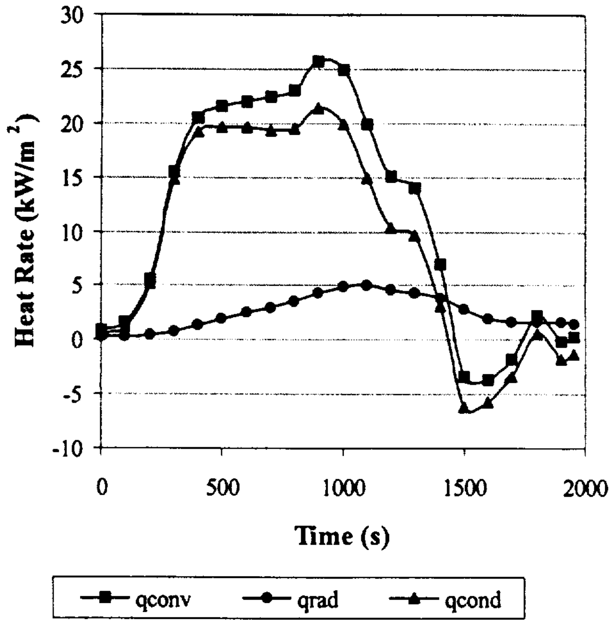

Figure 21. Leeward Body Point Heating History Case 2.

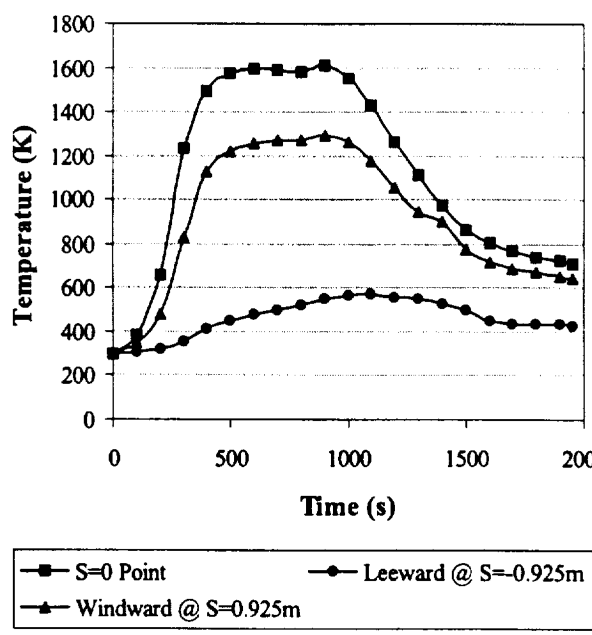

Figure 22. Temperature History Comparison.

Case 3: Stack Capability

One of the most important attributes of TCAT is its ability to analyze several disparate TPS materials sandwiched together as a stack. TCAT can analyze a range of two to five TPS materials layered together at a time. Figure 23 shows a representative five layer TPS stack. This stack is not intended to represent any TPS arrangements that are used in standard industry applications. It is intended as an example only.

The surface convective heat rate vs. time array obtained from the STS-1 trajectory analyzed in MINIVER was applied to the stack shown in Figure 23. This analysis was conducted at the point on the vehicle where the running length is defined as zero. Table 4 shows the thickness and number of nodes used for each of the materials.

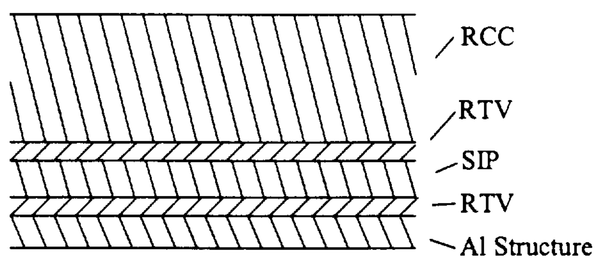

Figure 23. Schematic of TPS Stack.

Numerically simulated temperature profiles for the surface, RCC/RTV interface, and an aluminum (Al) structure backface are shown in Figure 24. The figure shows that the stack served its purpose of thermally insulates the Al structure from the severe flow field seen by the surface of the RCC. The temperature of the structure remained almost constant at 300 K while the surface attained a maximum value of approximately 1600 K.

Table 4. Material Thickness and Number of Nodes.

Material	*Thickness (mm)*	*No. of Nodes*
RCC	152.4	10
RTV	2.0	3
SIP	4.0	3
RTV	2.0	3
Al-Structure	25.4	5

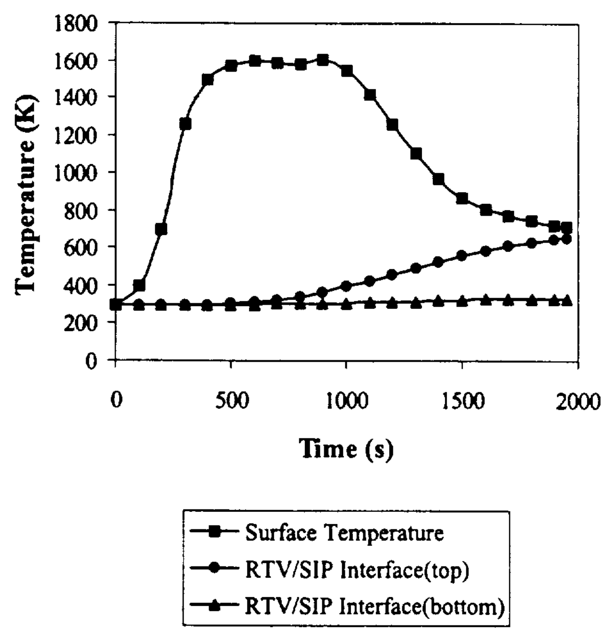

Figure 24: Temperature Distribution for TPS Stack of Five Materials.

<u>Benchmarking TCAT With SINDA</u>

Benchmark solutions for TCAT were obtained using the commercial software code SINDA. It is a thermal software package created by Network Analysis Associates, and it is capable solving lumped parameter representations of physical problems governed by diffusion type equations such as the heat equation.

Figure 25 shows a comparison of the SINDA and TCAT results for the transient-based analysis of the RCC tile located at S = 0 on a 37° half-angle cone flown along the STS-1 reentry trajectory. In addition, Figure 26 shows the results for the benchmark solution of the transient trajectory heating analysis of the TPS stack as shown in Figure 23. As can be seen, the SINDA and TCAT results match reasonably well. Therefore, this gives reassuring confidence in the solution methodology used in TCAT. Also, benchmarking of the TCAT solutions provides a proof of concept for an analysis capability that will be integrated into dynamic TPS design strategy.

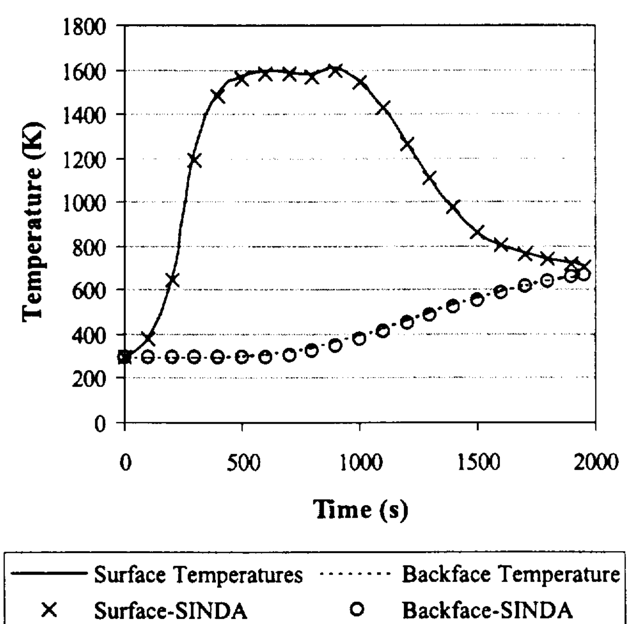

Figure 25. Benchmark Comparison of TCAT for RCC Tile on 37° Half-Angle Cone.

Another benchmark solution using SINDA was conducted in order to measure TCAT's performance when used in the dynamic TPS design strategy. TCAT was used to size materials on the 10° half-angle cone shown in Figure 27. The cone was flown along

the STS-1 reentry trajectory and analyzed using MINIVER in order to obtain the convective heat rate vs. time array for each body point in the MINIVER geometry file.

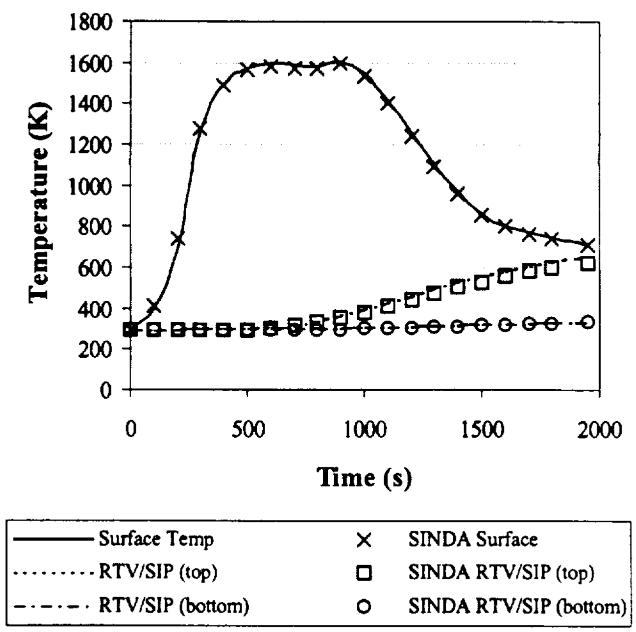

Figure 26. Benchmark Solution Comparison of TCAT for Five Material TPS Stack.

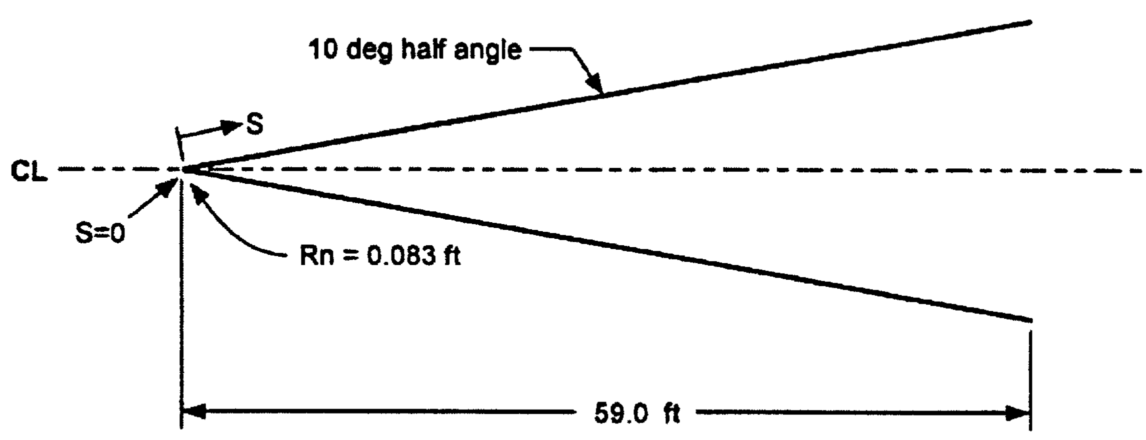

Figure 27. Schematic of 10° Half-Angle Cone.

The materials sized by TCAT were LI-900 tiles and AFRSI blankets. The blankets were positioned on the leeward side of the cone, and the tiles were placed on the windward. A total of 60 body points spaced two feet apart along the center mold line of the 10° cone were defined on the geometric model created in MINIVER. This resulted in 30 points on the leeward and windward sides of the cone.

Due to time constraints, only a total of 15 body points were analyzed in the SINDA heating analysis. Ten points on the leeward side of the cone were sized with AFRSI blankets, and five points sized with LI-900 tiles on the windward side. Also, the SINDA solution used the same trajectory, model geometry, TPS material properties, and heat rates from MINIVER as the TCAT solution.

The thermal responses of the two heating analyses were compared along with the differences of the average TPS unit weight values. The definition for the average TPS unit weight is given by equation (14).

$$average\ TPS\ unit\ weight = (average\ TPS\ thickness)(TPS\ density) \tag{14}$$

Table 5 shows the values obtained for the average TPS unit weights for the LI-900 tiles and AFRSI blankets calculated by TCAT and SINDA. As can be seen, the calculated average unit weights from TCAT and SINDA were very similar. The unit weight for the LI-900 tiles calculated by TCAT only differed by 1.7% from the values calculated by SINDA, and the AFRSI unit weight only differed by 0.62%!

Table 5. Comparison of TCAT and SINDA Average Unit Weights.

Material	*TCAT U.W. (lbm/ft²)*	*SINDA U.W. (lbm/ft²)*	*% Difference*
LI-900 tiles	1.558	1.532	1.70
AFRSI blankets	0.654	0.658	0.62

CHAPTER VII

TRADE OFF BETWEEN ACCURACY AND EXECUTION TIME

The previous chapter focused on proof of concept cases to demonstrate the capabilities of the TCAT heating tool. This chapter emphasizes the trade off between numerical accuracy and execution time of TCAT. In order to determine this, a time step sweep and spatial step sweep analysis was conducted. The time step is the amount of time incremented between time level iterations in the solution of the heat equation, and the spatial step is the distance between nodes in the numerical discretization of the TPS material stack. The spatial step is directly controlled by changing the number of nodes within each TPS material. As the number of nodes is increased, the spatial step is decreased. This sweep analysis was performed on the windward and leeward sides of the 10° half-angle cone, Figure 27, at a point two feet from its tip.

The leeward side time step and spatial step sweeps were conducted using AFRSI blanket material. The time sweep range was from 1 to 200 seconds with calculations being conducted at temporal increments of 1, 2, 5, 10, 25, 50, 100, and 200 seconds. The spatial step was controlled by the number of nodes that were placed in the AFRSI blanket material, and ranged from 10 to 80 nodes. The TCAT solution whose temperature results agreed best with the SINDA results was selected as the reference condition in this study. The SINDA solution that was used for the comparison utilized the same heating information, trajectory, and body point information for the 10° half-angle cone as the TCAT combination solutions.

Tables 6–9 list the maximum relative percent error calculated for each of the combinations of time step and number of nodes considered. The percent errors in the tables represent the maximum error of the surface and backface temperatures of each test combination compared to the same temperatures of the reference combination. The

reference combination for the leeward side numerical analysis was a time step of 1 sec with 40 nodes within the AFRSI material. In addition to AFRSI, this stack also consisted of RTV and GrEx. RTV and GrEx were discretized with 10 nodal points for all of the possible combinations considered in the leeward numerical sweep analysis. The definitions of the maximum relative errors used to calculate the values in Tables 6–9 are given by equations (15) and (16), respectively.

$$surface\ rel\ error = \frac{\max\ (ref\ surf\ temp\ \text{-}\ test\ case\ surf\ temp)}{avg(ref\ surface\ temperature)} * 100 \tag{15}$$

$$backface\ relative\ error = \frac{\max\ (ref\ backface\ temp\ \text{-}\ test\ case\ backface\ temp)}{avg(ref\ backface\ temperature)} * 100 \tag{16}$$

The average reference temperatures for Tables 6 and 7 were 612.9 K and 383.9 K, respectively. These are the average temperatures that occurred on the top surface and backface of the AFRSI material stack for the time duration of the heating analysis.

It can be seen that the lowest relative error for the surface temperature comparison in Table 6 occurred for a time step of 2 seconds and 40 nodes. This indicates that for a 1 second increase in the time step the maximum deviation of the surface temperature profile from reference solution is 1.06 %. It is desired to find a combination that will yield accurate results within 5–10% of the reference solution and require low CPU times. Short execution times are desired so that rapid and accurate calculation results are returned for a TCAT analysis conducted over the World Wide Web. Therefore, the best combinations of time step and number of nodes for the leeward side analyses are the circled entries in Tables 6 and 7. It was found that circled values in the table have an

execution time of approximately 30 seconds per body point compared to three minutes for the reference solutions.

The asterisks in the tables indicate that a solution was not obtained for the considered combination. It is suspected that for each of the combinations with an asterisk, and in fact, for all of the combinations with 50 or more nodes, round-off errors (due to small Δx) were significant and negatively affected the results. This is evident by the nearly constant or increasing values of the relative error that occurred as the time step and the number of nodes were increased. Thus, the runs with 50 or more nodes on the leeward side analysis are suspect due to round-off. This problem could be alleviated by non-dimensionalizing the spatial terms in the governing equations prior to numerical solution.

Table 6. Maximum Relative Error for Leeward Point Surface Temperatures.

	Number of Nodes							
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**
1	9.5 %	5.5 %	4.1 %	Reference	38.3 %	36.4 %	34.8 %	33.7 %
2	10.3 %	6.0 %	4.6 %	1.0 %	37.6 %	35.6 %	*	*
5	12.6 %	7.3 %	6.0 %	16.5 %	*	*	*	*
10	16.5 %	11.7 %	10.5 %	9.2 %	*	*	*	*
25	25.8 %	22.9 %	22.0 %	24.3 %	*	*	*	*
50	42.7 %	40.0 %	39.1 %	39.4 %	*	*	*	*
100	81.7 %	82.7 %	84.0 %	87.8 %	66.9 %	66.1 %	*	*
200	97.2 %	97.1 %	96.6 %	96.4 %	92.9 %	92.4 %	91.9 %	91.9 %

Suspected Round-Off Effect

Table 7. Maximum Relative Error for Leeward Point Backface Temperatures.

Time Step, seconds	Number of Nodes 10	20	30	40	50	60	70	80
1	10.2 %	8.5 %	5.0 %	Reference	6.4 %	9.7 %	12.6 %	14.9 %
2	10.2 %	8.5 %	5.0 %	0.04 %	6.4 %	9.7 %	*	*
5	10.1 %	8.4 %	4.9 %	9.9 %	*	*	*	*
10	9.9 %	8.3 %	4.8 %	0.14 %	*	*	*	*
25	9.6 %	7.9 %	4.6 %	0.40%	*	*	*	*
50	9.0 %	7.4 %	4.3 %	0.66 %	*	*	*	*
100	7.0 %	5.5 %	3.2 %	3.3 %	5.8 %	9.6 %	*	*
200	3.7 %	2.6 %	1.0 %	1.5 %	3.5 %	5.7 %	11.7 %	11.7 %

Suspected Round-Off Effect

The maximum values for the relative error of the time step and nodal number sweep analysis conducted on the windward side of the cone are given in Tables 8 and 9. The reference solution for the windward side analysis was selected in the same manner described for the leeward side analysis. Also, the windward side LI-900 tile TPS stack had four other material layers below the top tile material. These included two layers of RTV adhesive, a strain isolator pad, and the GrEx backface material. Each of these had 10 nodes that were held constant for each of the combinations that were analyzed in Tables 8 and 9. Finally, considering CPU time and accuracy tradeoffs, the best combinations of time step and number of nodes are indicated by the circled elements.

The combination of 40 nodes with a time step of 10 seconds was selected for leeward side WWW applications, and a combination of 40 nodes and a 5 second time step was selected for windward side applications.

Table 8. Maximum Relative Error for Windward Point Surface Temperatures.

Time Step, seconds	Number of Nodes 10	20	30	40	50	60
1	12.40 %	6.77 %	3.55 %	1.93 %	0.79 %	Reference
2	14.51 %	10.30 %	7.54 %	5.93 %	4.98 %	4.40 %
5	15.86 %	10.55 %	9.74 %	9.45 %	9.29 %	9.14 %
10	22.13 %	19.58 %	19.02 %	18.82 %	18.72 %	18.60 %
25	43.26 %	41.55 %	41.19 %	41.07 %	40.99 %	40.89 %
50	67.09 %	67.04 %	67.03 %	67.03 %	*	*

Table 9. Maximum Relative Error for Windward Point Backface Temperatures.

	Number of Nodes 10	20	30	40	50	60
1	8.71 %	8.15 %	6.83 %	5.16 %	2.95 %	Reference
2	8.70 %	8.14 %	6.82 %	5.17 %	2.96 %	0.01 %
5	8.62 %	8.06 %	6.76 %	5.13 %	2.97 %	0.06 %
10	8.58 %	8.02 %	6.74 %	5.13 %	3.01 %	0.12 %
25	8.48 %	7.93 %	6.68 %	5.12 %	3.14 %	0.36 %
50	8.31 %	7.76 %	6.58 %	6.58 %	*	*

CHAPTER VIII

WORLD WIDE WEB INTERFACE

Software Coupling

The motivation for the dynamic TPS sizing solution strategy is to have TCAT, ADS, and TPS*X* coupled in order to conduct the heating analysis and dynamic TPS sizing strategy via a WWW interface. This was accomplished using hyper-text-markup-language (HTML) and common-gateway-interface (CGI) scripting. The HTML and CGI programming was done on a UNIX platform.

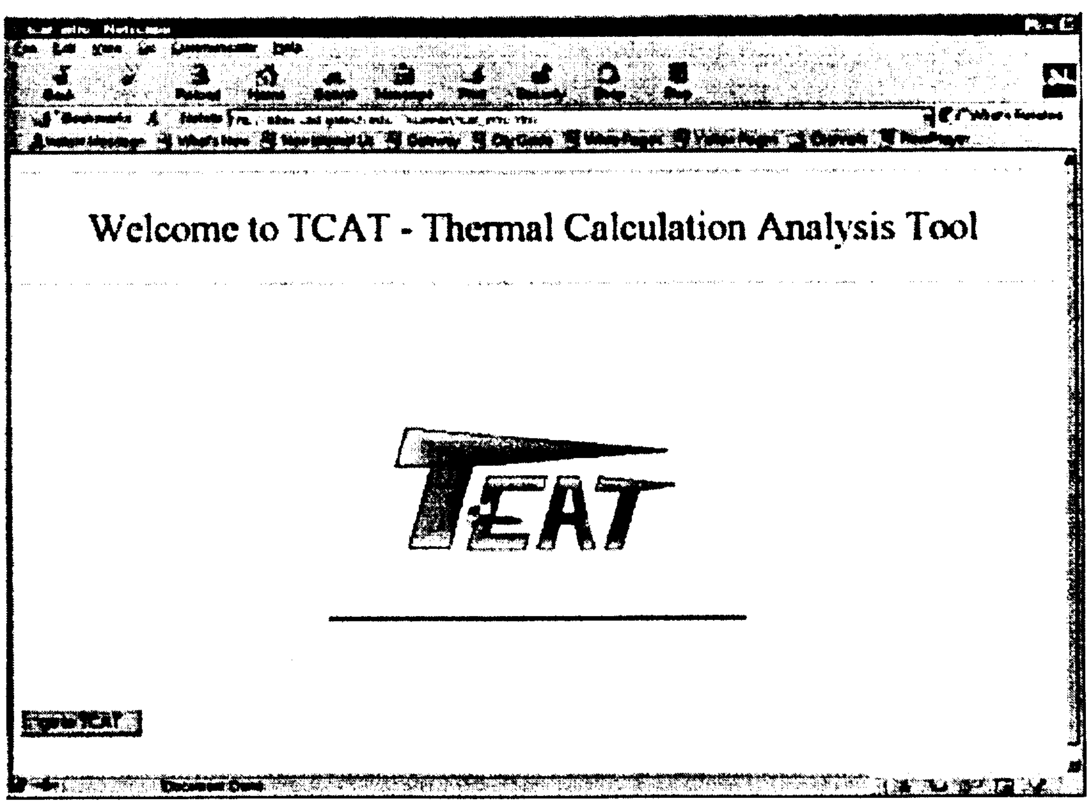

Figure 28. Screen Capture of TCAT Greeting.

At the initiation of the interface, the user is greeted by an animated GIF for the TCAT program, shown in Figure 28. After going through the greeting, the user views the main working environment of the web interface. The interface window as seen in Figure 29 is subdivided into three different frames: a header at the top of the window with input and output windows on the lower left and right hand sides, respectively. The input window provides the user with six different TPS design options. The first three involve design options of TPS systems that include several materials for the fuselage, cowl, and wing of a vehicle. The three remaining options allow the user to chose a particular TPS material and size it for the fuselage, cowl, and wing of a vehicle. Inputs and outputs listings for each of the design options are listed in Tables 10-15.

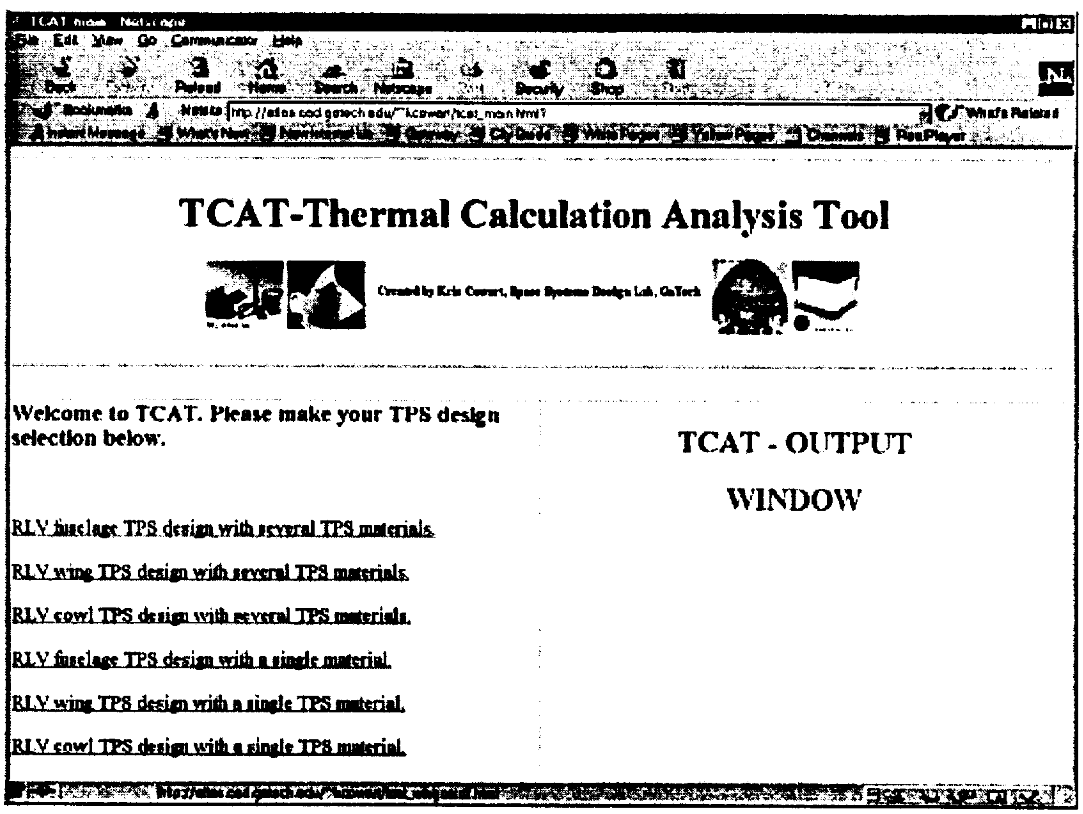

Figure 29. Screen Capture of TCAT Working Environment.

The next two subsections discuss the processes carried out by the scripts written for the automated Aeroheating analysis being discussed. The first subsection will discuss the details of the CGI scripts written for the TPS design involving several materials and the second will discuss the TPS design with a single material chosen by the designer.

Table 10. Inputs and Outputs for Fuselage TPS Design with Several Materials.

Inputs	*Outputs*
Approximate wetted body area	Thickness and material at each body point
Area % dominated by windward CL	Average unit weight for all materials
Area % dominated by leeward CL	Weight of SHARP material on nose
Backface material (GrEx or TiAl)	Average unit weight of each material
Windward or Leeward side	Ratio of each TPS material to fuselage
SHARP material on nose (Yes or No)	
TPS material family	
Name of MINIVER output file to be used	

Table 11. Inputs and Outputs for Wing TPS Design with Several Materials.

Inputs	*Outputs*
Approximate wetted wing area	Thickness and material at each body point
Area % dominated by windward CL	Average unit weight for all materials
Area % dominated by leeward CL	Weight/length of SHARP on leading
Backface material (GrEx or TiAl)	Average unit weight of each material
Windward or Leeward side	Ratio of each TPS material to wing area
SHARP material on wing (Yes or No)	
TPS material family	
Name of MINIVER output file to be used	

Table 12. Inputs and Outputs for Cowl TPS Design with Several Materials.

Inputs	*Outputs*
Approximate wetted cowl area	Thickness and material at each body point
Area % dominated by windward CL	Average unit weight for all materials
Area % dominated by leeward CL	Weight of SHARP on cowl leading edges
Backface material (GrEx or TiAl)	Average unit weight of each material
Windward or Leeward side	Ratio of each TPS material to cowl area
SHARP material on cowl (Yes or No)	
TPS material family	
Name of MINIVER output file to be used	

Table 13. Inputs and Outputs for Fuselage TPS Design with One Material.

Inputs	*Outputs*
Approximate wetted body area	Thickness at each body point
Area % dominated by windward CL	Weight of SHARP material on nose
Area % dominated by leeward CL	Average unit weight of material
Backface material (GrEx or TiAl)	Ratio of TPS material to fuselage area
Windward or Leeward side	
SHARP material on nose (Yes or No)	
TPS material to be used	
Name of MINIVER output file to be used	

Table 14. Inputs and Outputs for Wing TPS Design with One Material.

Inputs	*Outputs*
Approximate wetted wing area	Thickness at each body point
Area % dominated by windward CL	Weight/length of SHARP on lead edges
Area % dominated by leeward CL	Average unit weight of material
Backface material (GrEx or TiAl)	Ratio of TPS material to wing area
Windward or Leeward side	
SHARP material wing (Yes or No)	
TPS material to be used	
Name of MINIVER output file to be used	

Table 15. Inputs and Outputs for Cowl TPS Design with One Material.

Inputs	*Outputs*
Approximate wetted cowl area	Thickness at each body point
Area % dominated by windward CL	Weight of SHARP on cowl leading edge
Area % dominated by leeward CL	Average unit weight of material
Backface material (GrEx or TiAl)	Ratio of TPS material to cowl area
Windward or Leeward side	
SHARP material on cowl (Yes or No)	
TPS material to be used	
Name of MINIVER output file to be used	

Multiple Material Design Script

The first part of this script obtains the information from the website input frame and parses the information into the mesh type variable "$FORM{$name}." Next, the script goes through the MINIVER output file and creates individual files for each of the body points that contain time, convective heat rate, and radiation equilibrium temperature. This is accomplished by using an until-end-of-file-loop that searches line-by-line of the short version of the MINIVER output file. The MINIVER output file contains blocks of data for each of the points defined in the MINIVER input deck. Each

block of data begins with a line of text followed by columns of data that include information such as time, heat rate, heat load, etc., and ends with a "-1" flag. The until-end-of-file-loop starts at the text line, and parses through the columns of data until the "-1" flag is reached. This process is done for all of the body point blocks until the end of the MINIVER output file is reached.

Once this is completed, the script determines the TPS material to be used at a particular body point based on the radiation equilibrium temperature. This is accomplished by a "for" loop with nested if/then conditional statements. The "for" loop marches through the body points, and the conditional statements determine the material to be used based on temperature limits. In order to prevent a patchwork of materials from being chosen, an intelligent system had to be created. This was accomplished by looking at three different body points at a time and determining that the materials used were the same for those three points. If the materials on the two ends were different from the one in the middle and the middle material had a higher temperature limit, then the CGI script would switch the outer materials to that of the middle one. If the materials on the ends have higher temperature limits, then the middle material is changed to match those on the ends. After the materials are determined, the heating analysis is conducted. This is completed by a "for" loop using system-level calls that execute the FORTRAN code written for the heating analysis. After the heating analysis is completed, values for material thicknesses, unit weights, and acreage percentages are determined. These are in turn printed to the output window of the user web-interface.

Single Material Design Script

The process for a single TPS material is the same as that for several materials except that the CGI script does not determine the TPS material used. Instead, the user determines the material to be used as an input on the web-interface.

TPS Materials Selection

It was mentioned that the TPS materials for the multiple TPS material design option are selected by the CGI script based on the radiation equilibrium temperature obtained from MINIVER. These materials are segregated into three different groups: Shuttle technology materials, next generation RLV materials, and a combination of the two groups. Tables 16 – 18 list the materials in each of the groups. The user supplies the TPS material in the single material design options, Table 19. Also, the user has the option of selecting the type of backface material that is used in the TPS sizing analysis. There are two material options the user can choose from for the backface: graphite epoxy and titanium aluminide.

It is important to note that the next generation RLV materials are currently under development; therefore, specific information (i.e. material properties and unit weights) is restricted and cannot be published at this time. It can only be mentioned that such materials are included in design options for a restricted version of TCAT. On the other hand, information about the Shuttle technology materials group is not restricted, and is available for use on the unrestricted WWW version of TCAT.

Table 16. Shuttle Technology Materials.

Windward Materials	*Leeward Materials*
RCC tiles	FRCI tiles
LI-2200 tiles	AFRSI blankets
FRCI tiles	FRSI blankets

Table 17. Next Generation RLV Materials.

Windward Materials	*Leeward Materials*
RCC tiles	CFBI blankets
SiC tiles	AFRSI-2500 blankets
TUFI tiles	AFRSI-2200 blankets
	DURAFRSI blankets

Table 18. Group Combination of Materials.

Windward Materials	*Leeward Materials*
RCC tiles	CFBI blankets
SiC tiles	AFRSI-2500 blankets
AETB-12 tiles	AFRSI-2200 blankets
AETB-8 tiles	DURAFRSI blankets
LI-900 tiles	PBI blankets

Table 19. Options for Single TPS Material Design.

Windward Materials	*Leeward Materials*
AETB-8 tiles	AFRSI blankets
AETB-12 tiles	CFBI blankets
FRCI tiles	DURAFRSI blankets
FRCI 20 tiles	PBI blankets
LI-900 tiles	TABI blankets
LI-2200 tiles	
RCC tiles	
SiC tiles	
TUFI tiles	

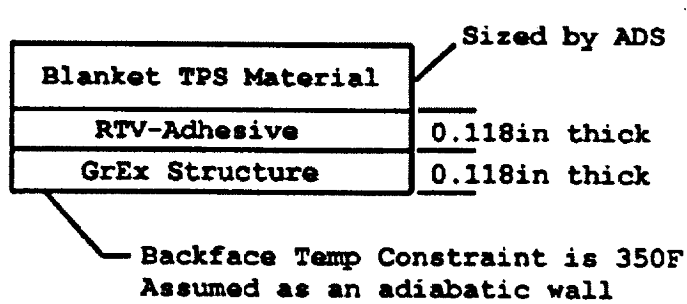

Figure 30. Schematic for Three Material Stack.

Materials used are either blankets or tiles. Each material is modeled as a TPS material stackup where the material chosen is the one that is sized. The blanket materials consist of a three layer stackup that includes the blanket insulation, an adhesive, and the backface material. Five materials make up tile configuration: a tile, an adhesive, a strain isolator pad, an adhesive, and the backface material. Figures 30 and 31 show schematics.

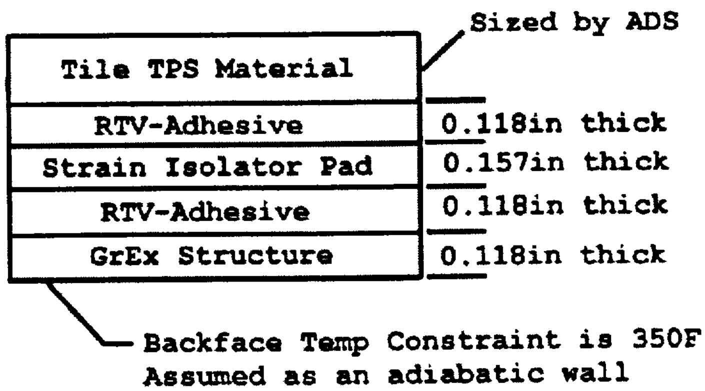

Figure 31. Schematic for Five Material Stack.

TUFI tiles are not modeled like those shown in Figure 31. Instead, they are modeled as laminates because of their high density, which can lead to undesirably high unit weights. Therefore, TUFI is assumed to maintain a constant thickness while AETB-8, placed underneath, is sized by ADS. Figure 32 shows this arrangement.

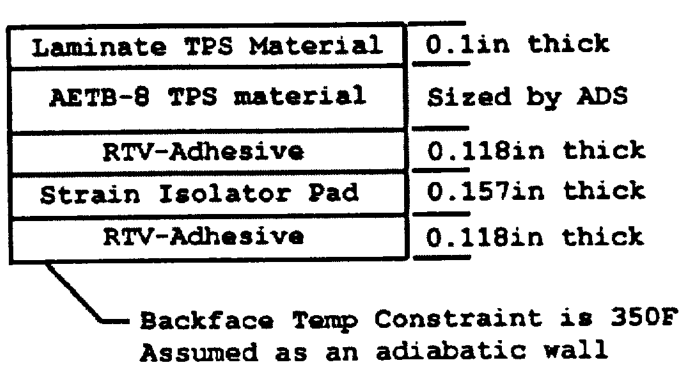

Figure 32. Schematic for Laminate Materials.

CHAPTER IX

DYNAMIC TPS DESIGN STRATEGY RESULTS AND APPLICATIONS

Single Material Design

A 10° half-angle spherical cone was chosen in order to demonstrate and test the user interface created for the one material TPS design option. The chosen trajectory was again the STS-1 reentry trajectory used earlier in Chapter VI. Figure 33 shows the geometry used to create the MINIVER input deck. Body points were placed two feet apart on the surface of the cone resulting in 30 body points for both the leeward and windward sides. The TPS materials of choice were AFRSI, Advanced Flexible Reusable Surface Insulation, blankets for the leeward surface and LI-900, 9 lb/ft^3 ceramic tiles, for the windward side. These two materials are shown in Figures 34 and 35 with inputs for each of the analyses shown in Tables 20 and 21.

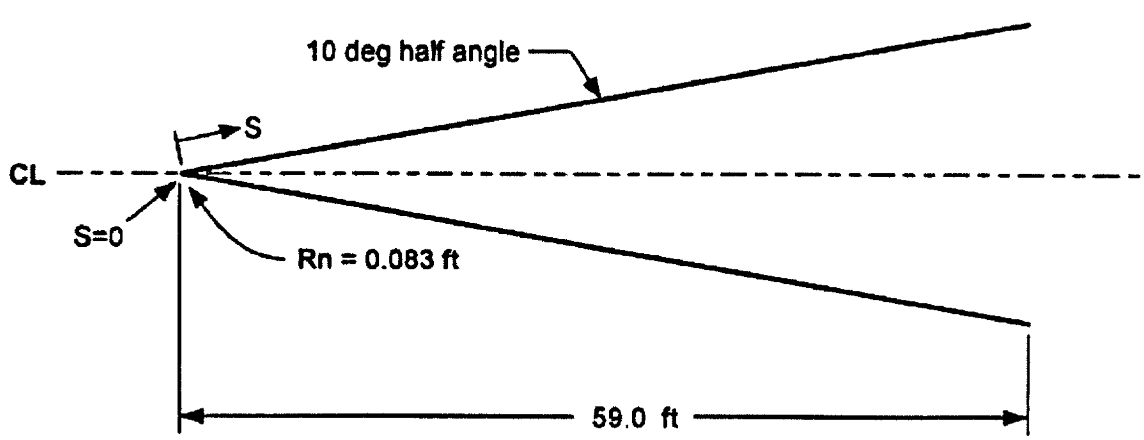

Figure 33. Schematic of 10° Half-Angle Cone.

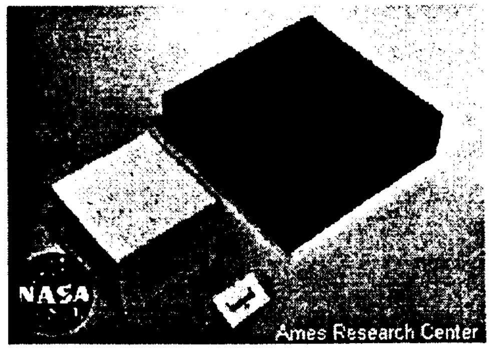

Figure 34. LI-900 Tile Sample.[2]

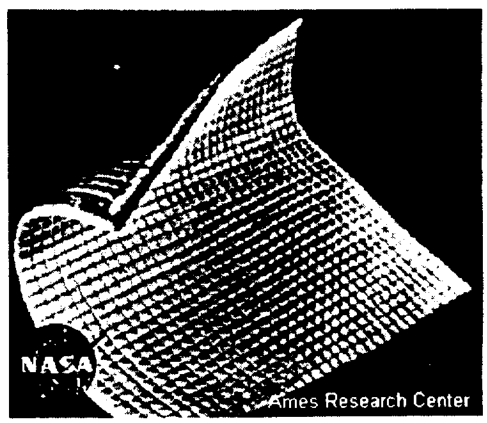

Figure 35. AFRSI Blanket Sample.[2]

Table 20. Inputs for Windward Side Analysis.

Input	*Value*
Surface area	1964.2 ft^2
Percent of area dominated by windward centerline	0.33
SHARP on nose	YES
TPS Material	LI-900
Backface Material	Graphite Epoxy

Table 21. Inputs for Leeward Side Analysis.

Input	*Value*
Surface area	1964.2 ft^2
Percent of area dominated by leeward centerline	0.67
SHARP on nose	YES
TPS Material	AFRSI
Backface Material	Graphite Epoxy

The heating analysis was conducted using TCAT in order to size the TPS materials at body points defined along the centerline of the leeward and windward sides of the cone. The calculated thickness values were then used to determine an average unit weight, equation (14) in Chapter VI, for each material of the TPS. The heating analysis was not conducted at the tip of the cone because TCAT is not capable of capturing 2-D heating effects. It was determined that SHARP materials were needed at the tip of the cone because according to MINIVER calculations the radiation equilibrium temperature exceeded 3500° F. SHARP materials are ultra high temperature ceramcs, such as hafnium diboride, that can withstand extreme temperatures resulting from high heating rates. These materials are under development by NASA Ames Research Center, and their design application is as a small radii leading edge, passive TPS for slender hypersonic vehicles. Hafnium diboride can withstand temperatures higher than the multi-use temperature limits of all other tile materials in the NASA Ames TPS*X* database. Arcjet test results have shown these materials can withstand temperatures in excess of (2480 K) 4000° F without material ablation.

Results of the TPS sizing for each side of the cone are given in Table 22. The unit weight obtained for the LI-900 tiles was approximately 1.42 lbm/ft^2 with thicknesses

ranging from 1.78 to 2.25 inches. A typical unit weight for windward side tiles is between 1.4 and 1.6 lbm/ft^2. The unit weight for the LI-900 tiles is therefore a good approximation of the average unit weight for the tiles on the windward side of the cone. The 0.5 lbm/ft^2 unit weight for the AFRSI blankets, includes a 0.3 lbm/ft^2 added areal weight. The desired range for the unit weight of leeward side blankets is from 0.4 to 0.6 lbm/ft^2. The thicknesses of the AFRSI blankets on the leeward side of the cone ranged from 0.25 to 1.68 inches.

Table 22. Output for 10° Half-angle Cone Heating Analysis.

Output	*Value*
LI-900 Unit Weight	1.42 lbm/ft^2
LI-900 Area to Body Area Ratio	0.33
LI-900 Average Thickness	1.9 in
AFRSI Unit Weight	0.50 lbm/ft^2
AFRSI Area to Body Area	0.67
AFRSI Average Thickness	0.40 in

Figure 36 shows examples of the iteration history of the LI-900 tile thickness for several body points on the windward side of the 10° half-angle cone. The locations of the body points are 2, 10, and 30 feet from the tip of the cone, respectively. The thickness of the each tile was controlled by the optimizer program ADS. ADS changed the thickness of the tiles at each of the body points in order to satisfy the temperature constraints that were set. As mentioned before, ADS is segmented into three levels: strategy, optimizer and one-dimensional search. The settings for each level used in this application and all others in this chapter are given in Table 23. As can be seen from Figure 36 the thickness

of the tile material at each of the body points converged within 4 to 5 iterations, and the thickness decreased for points further back on the cone.

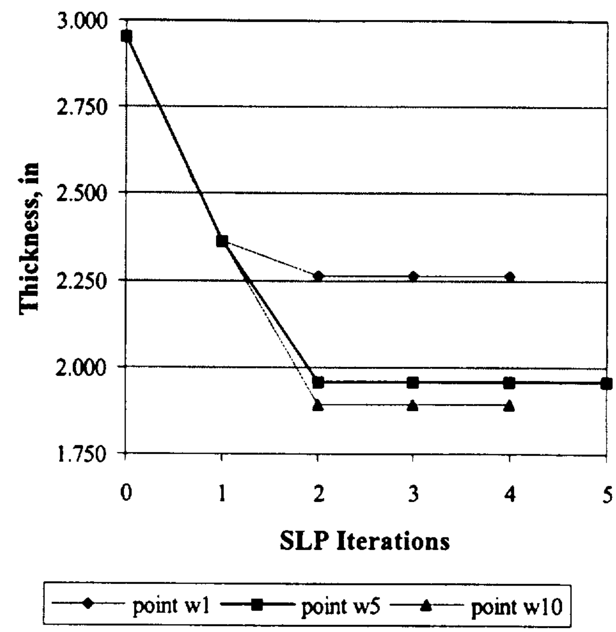

Figure 36. LI-900 Thickness Iteration History.

Table 23. ADS Settings.

ADS Operation Level	*Setting*
Strategy	Sequential Linear Programming
Optimizer	Modified Method of Feasible Directions
One Dimensional Search	Golden-Section Method

Multiple Material Designs

As discussed earlier, TPS materials for this design option are chosen from three different TPS material groups. These are Shuttle technology materials, next generation RLV materials, and a combination of the Shuttle technology and next generation RLV material groups. The 10° half-angle cone in Figure 33 and the five-foot wide wedge configuration in Figure 38 were flown along the STS-1 reentry trajectory and used to size TPS for the three material groups. Only results from the Shuttle technology materials group will be presented due to the sensitive nature of the next generation RLV and group combination TPS materials.

10° Half Angle Cone - Shuttle Technology Materials

The inputs for this analysis are the same as those for the single TPS material analysis except for the fact that the TPS materials were not input by the user. Instead, TCAT determined the TPS at each point based on the maximum radiation equilibrium temperature predicted by MINIVER. Results of the analysis showed that a combination of FRCI tiles, Fibrous Reinforced Composite Insulation, and LI-2200 tiles, a 22 lbm/ft^3 rigid ceramic tile, were used on the windward surface, and AFRSI blankets were used on the leeward surface of the cone.

Table 24 gives a detailed description of the output for the analysis of the 10° cone with the Shuttle technology materials. The thicknesses for the AFRSI blankets ranged from 0.25 to 1.65 inches. The LI-2200 tile thicknesses ranged from 1.45 to 1.89 inches, and the FRCI tiles were between 1.21 and 1.70 inches thick. Again, the AFRSI unit weight includes a 0.3 lbm/ft^2 additive areal weight. The unit weight of the AFRSI blankets is a reasonable estimate, but the values for the LI-2200 and FRCI tile unit weights are rather high in comparison to the target range discussed previously. The high unit weight value for the LI-2200 tiles most likely occurs since they roughly only account for 1% of the wetted TPS body area. This means that where the LI-2200 tiles are placed they are rather thick. Since their unit weight is based on the average thickness times the density of the material the unit weight will be high. In order to lower the unit weight

value for LI-2200 more points on the cone need to be covered with thinner LI-2200 tiles. The unit weight obtained for the FRCI tiles is more desirable, but is still quite high. The FRCI unit weight is high for the same reason given for the LI-2200 unit weight. It is important to mention that these materials are Shuttle technology. Once again, it is expected that more recent advancements in material technology will lead to lower unit weights for TPS materials. Figure 37 gives an illustration of the TPS layout for the Shuttle technology materials on the 10° half-angle cone.

Table 24. Shuttle Era TPS Materials on 10°Half-Angle Cone.

Output	***Value***
AFRSI Unit Weight	0.5 lbm/ft^2
AFRSI TPS Area to Body Area	0.67
AFRSI Average Thickness	0.4 in
LI-2200 Unit Weight	2.84 lbm/ft^2
LI-2200 TPS Area to Body Area	0.0990
LI-2200 Average Thickness	1.51 in
FRCI Unit Weight	1.43 lbm/ft^2
FRCI TPS Area to Body Area	0.2310
FRCI Average Thickness	1.44 in

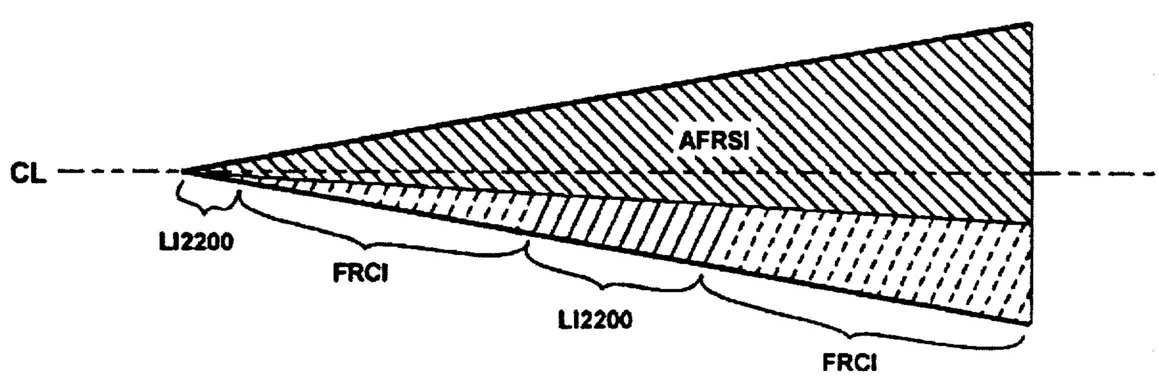

Figure 37. Shuttle Technology TPS Material Mapping on 10° Half-Angle Cone.

Multiple Angle Wedge - Shuttle Technology Materials

The multiple angle wedge configuration, shown in Figure 38, is a different analysis than that of the 10° half-angle cone. In this analysis, tangent wedge approximations were used instead of tangent cone approximations for the flow analysis conducted by MINIVER. Also, the body area percentages covered by blankets and tiles were different. The amount of surface area covered by tiles is 50%, with the same percentage for blankets. Further, there will be compressibility effects from the surface of the wedge due to the change in the flow angle over the surface of the wedge. The surface area used for the calculations was 909 ft^2 with SHARP materials are used on the leading edge of the nose due to the high radiation equilibrium temperatures at the tip of the wedge.

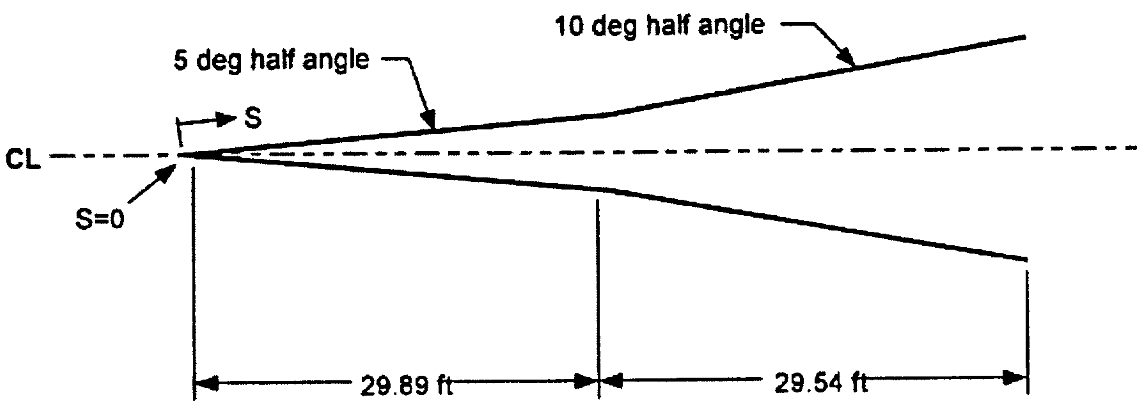

Figure 38. Schematic of Multiple Wedge Test Article.

Table 25 shows that AFRSI was selected as the material of choice for the leeward side of the wedge with a unit weight of 0.49 lbm/ft^2 including the 0.30 lbm/ft^2 added areal weight; the thickness for the AFRSI blankets ranged from 0.25 to 1.60 inches. The materials selected for the windward side of the wedge were LI-2200 tiles and FRCI tiles. There was a small area at the tip of the wedge that required LI-2200, and the recorded thickness was 1.89 in. The FRCI tile thicknesses ranged from 1.31 to 1.69 inches. The unit weight for the LI-900 material was relatively high at 3.46 lbm/ft^2. This is due to the fact that the area covered by LI-900 is only 0.17% of the total surface area. The FRCI

unit weight was 1.40 lbm/ft^2, which shows that if more surface area is available for the TPS material to occupy, then the unit weight will decrease. Figure 39 gives an illustration of the TPS layout for the Shuttle technology materials on the multiple angle wedge.

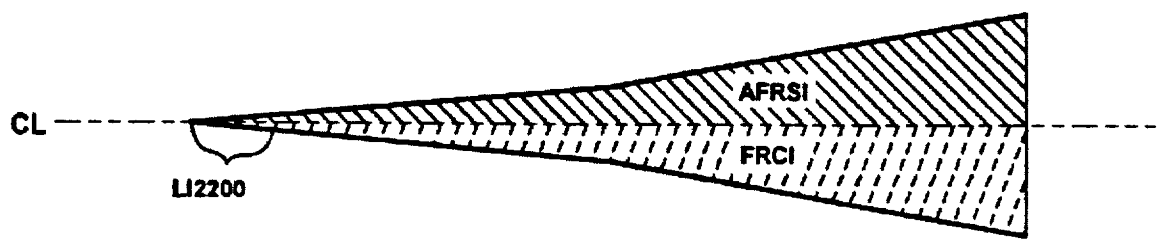

Figure 39. Shuttle Technology TPS Material Mapping on Multiple Angle Wedge.

Table 25. Shuttle Technology TPS Materials On Multiple Angle Wedge.

Output	*Value*
AFRSI Unit Weight	0.49 lbm/ft^2
AFRSI TPS Area to Body Area	0.5000
AFRSI Average Thickness	0.38 in
LI-2200 Unit Weight	3.46 lbm/ft^2
LI-2200 TPS Area to Body Area	0.0167
LI-2200 Average Thickness	1.89 in
FRCI Unit Weight	1.40 lbm/ft^2
FRCI TPS Area to Body Area	0.4833
FRCI Average Thickness	1.40 in

CHAPTER X

CONCLUSIONS AND RECOMMENDATIONS

This thesis presented the dynamic TPS design strategy to help integrate Aeroheating analysis and TPS sizing into a conceptual RLV design framework. This new strategy introduces feedback links into the design structure matrix enabling communication between Aeroheating and the other key RLV design disciplines (i.e., Aerodynamics and Trajectory). Also, the dynamic TPS sizing strategy allows sized TPS values to be revised and optimized during each iteration of an RLV design.

The integration of Aeroheating analysis involved the coupling of four design tools: TCAT, ADS, MINIVER, and TPS*X*. TCAT, the Thermal Calculation Analysis Tool, is an original code written for this research that uses finite difference methods coupled with optimization techniques in order to a conduct a transient, trajectory-based heating analysis to design and size the TPS of an RLV. The Automated Design Synthesis tool (ADS) is a software package that uses algorithms for the solution of constrained and unconstrained optimization problems. MINIVER is an analysis code that models the flowfield heating effects of important regions of an RLV. The Thermal Protection System Expert (TPS*X*) is a material properties database that is used for the selection of materials that will provide the thermal barrier insulation to the surface of an RLV.

Once these four computational tools were coupled together, a user interface was created in order to conduct the heating analysis and dynamic TPS sizing strategy via the World Wide Web. This was accomplished using hyper-text-markup-language (HTML) and common-gateway-interface (CGI) scripting. The HTML and CGI programming was done on a UNIX platform which provided the freedom of using the CGI scripts.

Results showing proof of concept for the heating code TCAT were also covered. These included a steady-state heating analysis given a constant heat rate applied to the

top surface of a tile. In addition, transient heating analyses capabilities based on a given trajectory were conducted. In addition, the ability to analyze a stack of disparate TPS materials was demonstrated. Benchmark solutions with the commercial heating code SINDA were performed and showed that results of the TCAT heating tool and dynamic TPS sizing strategy matched comparatively well.

A numerical analysis of the accuracy and execution time for the TCAT heating code was conducted by performing a sweeping analysis of the time step and spatial resolution. Results showed that the time required to perform the heating analysis at a single body point on a given geometry could be lowered from three minutes for an accurate solution to approximately 30 seconds with only a ten percent loss in accuracy.

Also, several applications demonstrating the performance of the automated dynamic TPS sizing strategy over the WWW were performed. Results showed that the TCAT tool can perform well as an acerage TPS design tool, and they proved the functionality of the TCAT tool in conceptual level RLV design.

There are some of recommendations if future work in this research is pursued.

1. The simple implicit method used to conduct the heating analysis in TCAT should be replaced with the Crank-Nicolson method. This is the same method used in the heating code SINDA. It is proven that the Crank-Niocolson method requires fewer nodes and less execution time. Along with this, it is numerically more accurate than the simple implicit scheme.
2. It is also recommended that a "problem specific optimizer" be written for the TPS sizing portion of the design strategy. ADS is a "general problem optimizer" in the sense that it was created to handle many different types of engineering design problems. It was found that ADS has many controlling parameters that have to be fine-tuned in order to achieve an optimal answer. A "problem specific optimizer" can alleviate this fine-tuning issue.

APPENDIX A

STS-1 REENTRY TRAJECTORY INFORMATION

Time (sec)	*Altitude (kft)*	*Velocity (kft/s)*	*Alpha (degrees)*	*Beta (degrees)*
45.3	373.8	24.6	41.3	0.0
90.3	351.5	24.6	41.2	0.0
135.3	329.5	24.6	40.5	0.0
225.3	288.0	24.7	41.9	0.0
270.3	269.7	24.6	39.4	0.0
315.3	256.3	24.5	40.7	0.0
360.3	250.2	24.2	41.7	0.0
405.3	247.0	23.9	40.0	0.0
450.3	244.6	23.6	39.3	0.0
495.3	242.5	23.3	39.7	0.0
785.3	225.2	20.4	40.2	0.0
809.3	223.0	20.0	40.2	0.0
833.3	219.0	19.7	40.3	0.0
857.3	215.3	19.2	40.4	0.0
881.3	211.4	18.8	40.1	0.0
905.3	207.6	18.3	40.3	0.0
929.3	205.6	17.8	40.0	0.0
953.3	202.5	17.2	42.0	0.0
977.3	197.2	16.5	40.9	0.0
1025.0	187.2	15.1	39.9	0.0
1049.0	182.6	14.4	39.4	0.0
1073.0	179.6	13.6	39.5	0.0
1240.0	150.0	8.3	34.1	0.0
1302.0	133.9	6.8	28.2	0.0
1364.0	117.4	5.3	23.1	0.0
1426.0	106.2	4.1	20.3	0.0
1488.0	89.1	2.9	16.5	0.0
1550.0	76.4	1.9	10.9	0.0
1612.0	57.4	1.2	7.8	0.0
1674.0	41.7	0.8	7.7	0.0
1736.0	27.6	0.7	7.1	0.0
1860.0	3.3	0.5	3.8	0.0

APPENDIX B

TCAT FORTRAN SOURCE CODE

```
      program tcat
c
c program to numerically optimize TPS unit weight for a TPS stack
c consisting of three dissparate TPS materials
c
c Kris Cowart, 17 Nov 99, updated 22 Nov 99
c
      implicit none
      integer nra,ncola,nrwk,nriwk,ndv,ncon,igrad,iwk(500),istrat,iopt,
     1ioned,iprint,info,i,idg(24),ic(24),n,j,matnum,nodes(6)
      real*8 x(12),vlb(12),vub(12),g(24),a(300,300),templmt(12)
      real*8 wk(1000),df(12),obj,ngt,epsi,initT,dt,time(5000),qcv(5000),
     1qrad(5000),qcond(5000),T(5000,100),k(6),den(6),cp(6),L(6)

c Also change in line above
       nra=300
       ncola=300
       nrwk=1000
       nriwk=500

c Problem setup 1 design variable, 6 constraints (2*number of materials)
      ndv=1
      ncon=10

c finite difference computed gradients
      igrad=0

c   set constraint types (g less than or equal to zero)
c   note: linear, inequality constraints should have idg = 2.
c         non-linear, inequality constraints should have idg = 0.
      do i=1,ncon
         idg(i)=2
      enddo
c
c read the initial guess for the thickness of the top TPS material
c
c obtain the istrat,iopt,ioned, and iprint info from input datafile
c
c read in the inputs from file
c

      open(10,file='inputs.in')
      read(10,*) matnum         !/number of materials
      read(10,*) dt             !/time step
      do i=1,matnum
      read(10,*) nodes(i)       !/number of nodes per material
      enddo
      read(10,*) epsi           !/emmissivity of surface
      do i=1,matnum
      read(10,*) k(i)           !/thermal conductivities
      enddo
      do i=1,matnum
      read(10,*) den(i)         !/densities
      enddo
      do i=1,matnum
      read(10,*) cp(i)          !/specific heats
      enddo
      read(10,*) initT          !/initial temperatures
      do i=1,matnum
      read(10,*) L(i)           !/thicknesses
```

```
      enddo
      do i=1,2*matnum
      read(10,*) templmt(i)     !/temp limits for each material
      enddo
      read(10,*) istrat         !/strategy of optimizer
      read(10,*) iopt           !/optimizer chosen
      read(10,*) ioned          !/one-d search method
      read(10,*) iprint         !/print format
      close(10)

      open(9,file='material_density')
      write(9,*) den(1)
      close(9)

c
c set the value of the design variable; the thickness of the top TPS material
c
      x(1)=L(1)

c
c set upper and lower boundaries on the design variable
c
      vlb(1)=0.00635      !/ thickness in meters (0.25 inches)
      vub(1)=0.241303     !/ thickness in meters (9.5 inches)
c
c call ads to initialize
c
c     info=0        !/ use ads default parameters
      info=-2       !/ change ads default parameter

      call ads (info,istrat,iopt,ioned,iprint,igrad,ndv,ncon,x,vlb,vub,
     1obj,g,idg,ngt,ic,df,a,nra,ncola,wk,nrwk,iwk,nriwk)

c     print*
c     print*,'info=',info
c     print*
c     pause

c
c   Change ADS default parameters if necessary
c   Note: many settings are problem specific. Check several variations.
c
c    set ctlmin to .0001 for linear constraints (allowable violation)
        wk(5) = .0001
c
c     set ctmin to .001 for non-linear constraints (allowable violation)
        wk(6) = .001
c
c  turn off automatic scaling
c
c        iwk(2)=0
c

c  reduce modified method of feasible directions push off factor
        wk(35) = 0.001
c
c  change relative and minimum absolute finite difference step size
        wk(21) = 1.e-4
        wk(22) = 1.e-4
c
```

```
c  change absolute and relative convergence criteria (optimizer-level)
         wk(8) = 1.e-3
         wk(12)= 1.e-4
c
c  change absolute and relative convergence criteria (strategy-level)
         wk(10)= 1.e-3
         wk(14)= 1.e-4
c
c change the maximum relative and absolute stepsize of the design variable
c taken in the first step of the one-d search

      wk(17)=1.e-3
      wk(18)=1e-4

c
c   change convergence criteria for Golden Section (if used)
         wk(7) = 1.e-5
         wk(11) = 1.e-6
c
c   change machine zero
         wk(37) = 1.e-10
c
c   increase maximum number of iterations
         iwk(3) = 100
         iwk(7) = 100

c
c Main Optimization Loop
c

 30   call ads (info,istrat,iopt,ioned,iprint,igrad,ndv,ncon,x,vlb,vub,
     1obj,g,idg,ngt,ic,df,a,nra,ncola,wk,nrwk,iwk,nriwk)

      if (info.eq.0) then
         write(*,*) ' optimization completed'
        write(91,2)
        do j=1,n
           write(91,4)time(j),qcv(j),qrad(j),qcond(j)
        enddo
        if(matnum.eq.1)then
           write(90,1)
           do j=1,n
              write(90,3)time(j),T(j,1),T(j,nodes(1))
           enddo
        elseif(matnum.eq.2)then
           write(90,5)
           do j=1,n
              write(90,6)time(j),T(j,1),T(j,nodes(1)),T(j,nodes(1)
     1+nodes(2))
           enddo
        elseif(matnum.eq.3)then
           write(90,7)
           do j=1,n
              write(90,8)time(j),T(j,1),T(j,nodes(1)),T(j,nodes(1)
     1+nodes(2)),T(j,nodes(1)+nodes(2)+nodes(3))
           enddo
        elseif(matnum.eq.4)then
           write(90,9)
           do j=1,n
              write(90,10)time(j),T(j,1),T(j,nodes(1)),T(j,nodes(1)
```

```
     1+nodes(2)),T(j,nodes(1)+nodes(2)+nodes(3)),T(j,nodes(1)+nodes(2)
     1+nodes(3)+nodes(4))
            enddo
         elseif(matnum.eq.5)then
            write(90,11)
            do j=1,n
               write(90,12)time(j),T(j,1),T(j,nodes(1)),T(j,nodes(1)
     1+nodes(2)),T(j,nodes(1)+nodes(2)+nodes(3)),T(j,nodes(1)+nodes(2)
     1+nodes(3)+nodes(4)),T(j,nodes(1)+nodes(2)+nodes(3)+nodes(4)+
     1nodes(5))
            enddo
         endif
         write(12,*)x(1)
          stop
        elseif (info.eq.1) then

           L(1)=x(1) !/ set the new value of the design variable

           if(matnum.eq.1)then
              call eval1(matnum,dt,nodes,epsi,k,den,cp,initT,L,templmt,obj
     1,x,g,time,qcv,qrad,qcond,T,n)
           elseif(matnum.eq.2)then
              call eval2(matnum,dt,nodes,epsi,k,den,cp,initT,L,templmt,obj
     1,x,g,time,qcv,qrad,qcond,T,n)
           elseif(matnum.eq.3)then
              call eval3(matnum,dt,nodes,epsi,k,den,cp,initT,L,templmt,obj
     1,x,g,time,qcv,qrad,qcond,T,n)
           elseif(matnum.eq.4)then
              call eval4(matnum,dt,nodes,epsi,k,den,cp,initT,L,templmt,obj
     1,x,g,time,qcv,qrad,qcond,T,n)
           elseif(matnum.eq.5)then
              call eval5(matnum,dt,nodes,epsi,k,den,cp,initT,L,templmt,obj
     1,x,g,time,qcv,qrad,qcond,T,n)
           endif
           goto 30
        else
          write(*,*) 'program stopped prematurely'
          stop
        endif
1       format(2x,'time(j)',7x,'T(j,1)',5x,'T(j,nodes(1))')
2       format(2x,'time',7x,'qconv',5x,'qrad',5x,'qcond')
3       format(f12.2,f12.2,f12.2)
4       format(f12.2,3x,f11.2,3x,f11.2,3x,f11.2)
5       format(5x,'time',5x,'T(j,1)',5x,'T(j,nodes1)',5x
     1,'T(j,totnodes)')
6       format(f12.2,f12.2,f12.2,f12.2)
7       format(2x,'time(j)',7x,'T(j,1)',5x,'T(j,nodes(1))',2x
     1,'T(j,sum1)',5x,'T(j,sum2)')
8       format(f8.2,7x,f8.2,5x,f8.2,4x,f8.2,5x,f8.2)
9       format(2x,'time(j)',7x,'T(j,1)',5x,'T(j,nodes(1))',2x
     1,'T(j,sum1)',5x,'T(j,sum2)',5x,'T(j,totnodes)')
10      format(f8.2,7x,f8.2,5x,f8.2,4x,f8.2,5x,f8.2,5x,f8.2)
11      format(2x,'time(j)',7x,'T(j,1)',5x,'T(j,nodes(1))',2x
     1,'T(j,sum1)',5x,'T(j,sum2)',5x,'T(j,sum3)',5x,'T(j,totnodes)')
12      format(f8.2,7x,f8.2,5x,f8.2,4x,f8.2,5x,f8.2,5x,f8.2,5x,f8.2)
          end  !/*** end of main code ***
```

```fortran
      subroutine evall(matnum,dt,nodes,epsi,k,den,cp,initT,L,templmt,obj
     1,x,g,time,qcv,qrad,qcond,T,n)

      implicit none
      integer nodes(6),totnodes,n,i,j,matnum
      real*8 x(12),g(24),obj,den1,templmt(12),T(5000,100),epsi,k(6)
     1,den(6),cp(6),initT,L(6),dt,diff(5000,24),maxdiff(12)
     1,time(5000),qcv(5000),qrad(5000),qcond(5000)

      obj = den(1)*x(1)

      call onestack(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n
     1,time,qcv,qcond,qrad)

      do j=1,n
         diff(j,1)=T(j,1)-templmt(1)
         diff(j,2)=T(j,nodes(1))-templmt(2)
             enddo
      do i=1,2*matnum
         maxdiff(i)=diff(1,i)
      enddo
      do j=2,n
         do i=1,2*matnum
            if(diff(j,i).gt.maxdiff(i))then
               maxdiff(i)=diff(j,i)
            endif
         enddo
      enddo
      do i=1,2*matnum
         g(i)=maxdiff(i)
      enddo
      do i=2*matnum+1,24
         g(i)=0.
      enddo
      write(11,1)x(1),obj,g(1),g(2),g(3),g(4)
1     format(3x,f12.6,3x,f12.6,3x,e12.6,3x,e12.6,3x,e12.6,3x,e12.6)
      return
      end
```

```
      subroutine eval2(matnum,dt,nodes,epsi,k,den,cp,initT,L,templmt,obj
     1,x,g,time,qcv,qrad,qcond,T,n)

      implicit none
      integer nodes(6),totnodes,n,i,j,matnum
      real*8 x(12),g(24),obj,den1,templmt(12),T(5000,100),epsi,k(6)
     1,den(6),cp(6),initT,L(6),dt,diff(5000,24),maxdiff(12)
     1,time(5000),qcv(5000),qrad(5000),qcond(5000)


      obj = den(1)*x(1)

      call twostack(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n
     1,time,qcv,qcond,qrad)

      totnodes=nodes(1)+nodes(2)
      do j=1,n
         diff(j,1)=T(j,1)-templmt(1)
         diff(j,2)=T(j,nodes(1))-templmt(2)
         diff(j,3)=T(j,nodes(1))-templmt(3)
         diff(j,4)=T(j,totnodes)-templmt(4)

      enddo
      do i=1,2*matnum
         maxdiff(i)=diff(1,i)
      enddo
      do j=2,n
         do i=1,2*matnum
            if(diff(j,i).gt.maxdiff(i))then
               maxdiff(i)=diff(j,i)
            endif
         enddo
      enddo
      do i=1,2*matnum
         g(i)=maxdiff(i)
      enddo
      do i=2*matnum+1,24
         g(i)=0.
      enddo
      write(11,1)x(1),obj,g(1),g(2),g(3),g(4)
1     format(3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6)
      return
      end
```

```
      subroutine eval3(matnum,dt,nodes,epsi,k,den,cp,initT,L,templmt,obj
     1,x,g,time,qcv,qrad,qcond,T,n)

      implicit none
      integer nodes(6),totnodes,n,i,j,matnum
      real*8 x(12),g(24),obj,den1,templmt(12),T(5000,100),epsi,k(6)
     1,den(6),cp(6),initT,L(6),dt,diff(5000,24),maxdiff(12)
     1,time(5000),qcv(5000),qrad(5000),qcond(5000)

      obj = den(1)*x(1)

      call threestack(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n
     1,time,qcv,qcond,qrad)

      totnodes=nodes(1)+nodes(2)+nodes(3)
      do j=1,n
         diff(j,1)=T(j,1)-templmt(1)
         diff(j,2)=T(j,nodes(1))-templmt(2)
         diff(j,3)=T(j,nodes(1))-templmt(3)
         diff(j,4)=T(j,nodes(1)+nodes(2))-templmt(4)
         diff(j,5)=T(j,nodes(1)+nodes(2))-templmt(5)
         diff(j,6)=T(j,totnodes)-templmt(6)

      enddo

c        do j=1,n
c        write(13,*) time(j),T(j,1),T(j,totnodes)
c        enddo

      do i=1,2*matnum
         maxdiff(i)=diff(1,i)
      enddo
      do j=2,n
         do i=1,2*matnum
            if(diff(j,i).gt.maxdiff(i))then
               maxdiff(i)=diff(j,i)
            endif
         enddo
      enddo
      do i=1,2*matnum
         g(i)=maxdiff(i)
      enddo
      do i=2*matnum+1,24
         g(i)=0.
      enddo
      write(11,1)x(1),obj,g(1),g(2),g(3),g(4),g(5),g(6)
 1    format(3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,
     13x,f12.6,3x,f12.6)
      return
      end
```

```
      subroutine eval4(matnum,dt,nodes,epsi,k,den,cp,initT,L,templmt,obj
     1,x,g,time,qcv,qrad,qcond,T,n)

      implicit none
      integer nodes(6),totnodes,n,i,j,matnum
      real*8 x(12),g(24),obj,templmt(12),T(5000,100),epsi,k(6)
     1,den(6),cp(6),initT,L(6),dt,diff(5000,24),maxdiff(12)
     1,time(5000),qcv(5000),qrad(5000),qcond(5000)

c
c the objection function for the optimizer is a function of density and the
c thickness of the top layer TPS material
c
      obj=den(1)*x(1)
c
c evaluate the heat transfer and temperature profiles for the constituient
c materials of the stack
c
      call fourstack(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n
     1,time,qcv,qcond,qrad)
c
c calculate the difference between the calculated temperature and the
c max allowable temperature limit from TPSX
c
      totnodes=nodes(1)+nodes(2)+nodes(3)+nodes(4)
      do j=1,n
         diff(j,1)=T(j,1)-templmt(1)
         diff(j,2)=T(j,nodes(1))-templmt(2)
         diff(j,3)=T(j,nodes(1))-templmt(3)
         diff(j,4)=T(j,nodes(1)+nodes(2))-templmt(4)
         diff(j,5)=T(j,nodes(1)+nodes(2))-templmt(5)
         diff(j,6)=T(j,nodes(1)+nodes(2)+nodes(3))-templmt(6)
         diff(j,7)=T(j,nodes(1)+nodes(2)+nodes(3))-templmt(7)
         diff(j,8)=T(j,totnodes)-templmt(8)
      enddo
      do i=1,2*matnum
         maxdiff(i)=diff(1,i)
      enddo
      do j=2,n
         do i=1,2*matnum
            if(diff(j,i).gt.maxdiff(i))then
               maxdiff(i)=diff(j,i)
            endif
         enddo
      enddo
      do i=1,2*matnum
         g(i)=maxdiff(i)
      enddo
      do i=2*matnum+1,24
         g(i)=0.
      enddo
      write(11,1)x(1),obj,g(1),g(2),g(3),g(4),g(5),g(6),g(7),g(8)
 1    format(3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,
     13x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6)
      return
      end
```

```
      subroutine eval5(matnum,dt,nodes,epsi,k,den,cp,initT,L,templmt,obj
     1,x,g,time,qcv,qrad,qcond,T,n)

      implicit none
      integer nodes(6),totnodes,n,i,j,matnum
      real*8 x(12),g(24),obj,templmt(12),T(5000,100),epsi,k(6)
     1,den(6),cp(6),initT,L(6),dt,diff(5000,24),maxdiff(12)
     1,time(5000),qcv(5000),qrad(5000),qcond(5000)

c
c the objection function for the optimizer is a function of density and the
c thickness of the top layer TPS material
c
      obj=den(1)*x(1)

c
c evaluate the heat transfer and temperature profiles for the constituient
c materials of the stack
c
      call fivestack(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n
     1,time,qcv,qcond,qrad)
c
c calculate the difference between the calculated temperature and the
c max allowable temperature limit from TPSX
c
      totnodes=nodes(1)+nodes(2)+nodes(3)+nodes(4)+nodes(5)
      do j=1,n
         diff(j,1)=T(j,1)-templmt(1)
         diff(j,2)=T(j,nodes(1))-templmt(2)
         diff(j,3)=T(j,nodes(1))-templmt(3)
         diff(j,4)=T(j,nodes(1)+nodes(2))-templmt(4)
         diff(j,5)=T(j,nodes(1)+nodes(2))-templmt(5)
         diff(j,6)=T(j,nodes(1)+nodes(2)+nodes(3))-templmt(6)
         diff(j,7)=T(j,nodes(1)+nodes(2)+nodes(3))-templmt(7)
         diff(j,8)=T(j,nodes(1)+nodes(2)+nodes(3)+nodes(4))-templmt(8)
         diff(j,9)=T(j,nodes(1)+nodes(2)+nodes(3)+nodes(4))-templmt(9)
         diff(j,10)=T(j,totnodes)-templmt(10)
      enddo
      do i=1,2*matnum
         maxdiff(i)=diff(1,i)
      enddo
      do j=2,n
         do i=1,2*matnum
            if(diff(j,i).gt.maxdiff(i))then
               maxdiff(i)=diff(j,i)
            endif
         enddo
      enddo
      do i=1,2*matnum
         g(i)=maxdiff(i)
      enddo
      do i=2*matnum+1,24
         g(i)=0.
      enddo
      write(11,1)x(1),obj,g(1),g(2),g(3),g(4),g(5),g(6),g(7),g(8),g(9),g(10)
 1    format(3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,
     13x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6,3x,f12.6)
      return
      end
```

```
      subroutine onestack(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n
     1,time,qcv,qcond,qrad)
c      program heattx
c
c this program will use the Newton Raphson Method to solve for the
c heat transfer to a point on a tile over time
c
      implicit none
      integer i,n,nodes(6),j,matnum
      real*8 initT,a(1000),b(1000),c(1000),delT(1000),T(5000,100)
     1,k(6),cp(6),sigma,epsi,dx(6),dt,L(6),x(1000),alpha(6),lam(6)
     1,eps,den(6),f(1000),time(5000),qcond(5000),qrad(5000),qcv(5000)
     1,Tradeq(5000),normdelT,normf,twonorm,dTdx(5000)

      external twonorm

      call linint(n,dt,time,qcv,Tradeq)

c
c definition of the constants
c
      sigma=5.67051e-8
      epsi=0.83
      dx(1)=L(1)/(nodes(1)-1.)
      x(1)=0.
      do i=2,nodes(1)
         x(i)=x(i-1)+dx(1)
      enddo
      alpha(1)=k(1)/(den(1)*cp(1)*1000.)
      lam(1)=alpha(1)*dt/(dx(1)**2)
c      print*,'alpha(1)=',alpha(1)
c      print*,'lambda(1)=',lam(1)
c
c apply the initial conditions
c
      do i=1,nodes(1)
         T(1,i)=initT
      enddo
c
c make the initial guess for the temperature at the next time step
c
      do i=1,nodes(1)
         T(2,i)=1000.
      enddo
c
c set up the Jacobian matrix for the system of equations
c
      do i=2,nodes(1)-1
         a(i)=lam(1)
         b(i)=-1.*(1.+2.*lam(1))
         c(i)=lam(1)
      enddo
      a(nodes(1))=0.
      c(1)=0.
      a(1)=2.*lam(1)
      b(1)=-2.*lam(1)-8.*(lam(1)*dx(1)*epsi*sigma*T(2,1)**3)/k(1)-1.
      b(nodes(1))=-1.*(1.+2.*lam(1))
      c(1)=2.*lam(1)
c
c Solution of the first time step
```

```
c
      time(1)=0.
      eps=1.e-3
      normf=1.
      normdelT=1.
      dowhile((normf.gt.eps).or.(normdelT.gt.eps))
         f(1)=-2.*lam(1)*(T(2,2)-T(2,1))+2.*lam(1)*dx(1)*((epsi*sigma*
     1T(1,1)**4-qcv(1))/k(1))+T(2,1)-T(1,1)
         f(nodes(1))=-2.*lam(1)*T(2,nodes(1)-1)+(1.+2.*lam(1))
     1*T(2,nodes(1))-T(1,nodes(1))
         do i=2,nodes(1)-1
            f(i)=-1.*lam(1)*T(2,i-1)+(1.+2.*lam(1))*T(2,i)-lam(1)*T(2,i+1)
     1-T(1,i)
         enddo

         call tridag(a,b,c,f,delT,nodes)

         do i=1,nodes(1)
            T(2,i)=T(2,i)+delT(i)
         enddo
         b(1)=-2.*lam(1)-8.*(lam(1)*dx(1)*epsi*sigma*T(2,1)**3)/k(1)-1.
c
c Use the Eucledian Norm to test convergence of each time step
c
         normf=twonorm(f,nodes)
         normdelT=twonorm(delT,nodes)
      enddo
      time(2)=time(1)+dt

c
c Include the remaining time levels
c
      do j=2,n
         do i=1,nodes(1)
            T(j+1,i)=T(j,i)
         enddo
         b(1)=-2.*lam(1)-4.*(lam(1)*dx(1)*epsi*sigma*T(j+1,1)**3)/k(1)-1.
         normf=1.
         normdelT=1.
         dowhile((normf.gt.eps).or.(normdelT.gt.eps))
            f(1)=-2.*lam(1)*(T(j+1,2)-T(j+1,1))+2.*lam(1)*dx(1)*
     +((epsi*sigma*T(j+1,1)**4-qcv(j))/k(1))+T(j+1,1)-T(j,1)
            f(nodes(1))=-2.*lam(1)*T(j+1,nodes(1)-1)+(1.+2.*lam(1))
     +*T(j+1,nodes(1))-T(j,nodes(1))
            do i=2,nodes(1)-1
               f(i)=-1.*lam(1)*T(j+1,i-1)+(1.+2.*lam(1))*T(j+1,i)-
     +lam(1)*T(j+1,i+1)-T(j,i)
            enddo

            call tridag(a,b,c,f,delT,nodes)

            do i=1,nodes(1)
               T(j+1,i)=T(j+1,i)+delT(i)
            enddo
            b(1)=-2.*lam(1)-8.*(lam(1)*dx(1)*epsi*sigma*
     1T(j+1,1)**3)/k(1)-1.
c
c Use the Eucledian Norm to test convergence of each time step
c
            normf=twonorm(f,nodes)
```

```
      normdelT=twonorm(delT,nodes)
   enddo
   time(j+1)=time(j)+dt
enddo
do j=1,n
   qrad(j)=epsi*sigma*T(j,1)**4
   qcond(j)=qcv(j)-qrad(j)
enddo
return
end
```

```
c program that numerically predicts the heat transfer into a stack of TPS
c materials.  this first case will investigate RCC on top of Aluminum.
c
c Kris Cowart 8 Oct 99
c
      subroutine twostack(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n
     1,time,qcv,qcond,qrad)

      implicit none
      integer i,j,n,nodes(6),totnodes,matnum
      real*8 dt,k(6),den(6),cp(6),initT,L(6),dx(6),sigma,epsi
      real*8 eps,alpha(6),lam(6),a(100),b(100),c(100),T(5000,100)
      real*8 normf,normdelT,qconv,f(100),delT(100),twonorm,qcv(5000)
      real*8 Tradeq(5000),time(5000),qcond(5000),qrad(5000)
      external twonorm

      call linint(n,dt,time,qcv,Tradeq)
c
c define the constants
c
      eps=1.e-6
      sigma=5.67061e-8
      do i=1,matnum
          dx(i)=L(i)/(nodes(i)-1.)
         alpha(i)=k(i)/(den(i)*cp(i)*1000.)
         lam(i)=alpha(i)*dt/(dx(i)**2)
      enddo
      totnodes=nodes(1)+nodes(2)
c
c apply the initial conditions
c
      do i=1,totnodes
         T(1,i)=initT
      enddo
c
c make the initial guess for the temperature at the next time step
c
      do i=1,totnodes
         T(2,i)=1000.
      enddo
c
c set up the Jacobian Matrix for the system of equations
c
      a(1)=-2.*lam(1)
      b(1)=1.+2.*lam(1)+(8.*lam(1)*dx(1)*epsi*sigma*T(2,1)**3)/k(1)
      c(1)=0.
      do i=2,nodes(1)-1
         a(i)=-1.*lam(1)
         b(i)=1.+2.*lam(1)
         c(i)=-1.*lam(1)
      enddo
      a(nodes(1))=-2.*lam(2)
      b(nodes(1))=1.+2.*lam(2)+2.*lam(2)*k(1)/k(2)
      c(nodes(1))=-2.*lam(2)*k(1)/k(2)
      do i=nodes(1)+1,totnodes-1
         a(i)=-1.*lam(2)
         b(i)=1.+2.*lam(2)
         c(i)=-1.*lam(2)
      enddo
      a(totnodes)=0.
```

```
      b(totnodes)=1.+2.*lam(2)
      c(totnodes)=-2.*lam(2)
c
c solution at the first time step
c
      normf=1.
      normdelT=1.
      time(1)=0.
      dowhile((normf.gt.eps).or.(normdelT.gt.eps))
         f(1)=2.*lam(1)*T(2,2)-T(2,1)-2.*lam(1)*T(2,1)-(2.*lam(1)*dx(1)
     1*epsi*sigma*T(2,1)**4)/k(1)+2.*lam(1)*dx(1)*qcv(1)/k(1)+T(1,1)
         do i=2,nodes(1)
            f(i)=lam(1)*T(2,i-1)-(1.+2.*lam(1))*T(2,i)+lam(1)
     1*T(2,i+1)+T(1,i)
         enddo
         f(nodes(1))=2.*lam(2)*T(2,nodes(1)+1)-(1.+2.*lam(2)
     1+2.*lam(2)*k(1)/k(2))*T(2,nodes(1))+2.*lam(2)*k(1)/k(2)
     1*T(2,nodes(1)-1)+T(1,nodes(1))
         do i=nodes(1)+1,totnodes-1
            f(i)=lam(2)*T(2,i-1)-(1.+2.*lam(2))*T(2,i)
     1+lam(2)*T(2,i+1)+T(1,i)
         enddo
         f(totnodes)=-1.*(1.+2.*lam(2))*T(2,totnodes)+2.*lam(2)
     1*T(2,totnodes-1)+T(1,totnodes)
c
c send inforamtion to the tridag solver
c
         call tridag(a,b,c,f,delT,totnodes)
c
c update temperature values
c
         do i=1,totnodes
            T(2,i)=T(2,i)+delT(i)
         enddo
         b(1)=1.+2.*lam(1)+(8.*lam(1)*dx(1)*epsi*sigma*T(2,1)**3)
     1/k(1)
         normf=twonorm(f,totnodes)
         normdelT=twonorm(delT,totnodes)
      enddo
      time(2)=time(1)+dt
c
c solution of system by marching in time
c
      do j=2,n
         do i=1,totnodes
            T(j+1,i)=T(j,i)
         enddo
         b(1)=1.+2.*lam(1)+(8.*lam(1)*dx(1)*epsi*sigma
     1*T(j+1,1)**3)/k(1)
         normf=1.
         normdelT=1.
         dowhile((normf.gt.eps).or.(normdelT.gt.eps))
            f(1)=2.*lam(1)*T(j+1,2)-T(j+1,1)-2.*lam(1)*T(j+1,1)
     1-(2.*lam(1)*dx(1)*epsi*sigma*T(j+1,1)**4)/k(1)+2.*lam(1)*dx(1)
     1*qcv(j)/k(1)+T(j,1)
            do i=2,nodes(1)
               f(i)=lam(1)*T(j+1,i-1)-(1.+2.*lam(1))*T(j+1,i)
     1+lam(1)*T(j+1,i+1)+T(j,i)
            enddo
            f(nodes(1))=2.*lam(2)*T(j+1,nodes(1)+1)-(1.
```

```
     1+2.*lam(2)+2.*lam(2)*k(1)/k(2))*T(j+1,nodes(1))
     1+2.*lam(2)*k(1)/k(2)*T(j+1,nodes(1)-1)+T(j,nodes(1))
            do i=nodes(1)+1,totnodes-1
               f(i)=lam(2)*T(j+1,i-1)-(1.+2.*lam(2))*T(j+1,i)
     1+lam(2)*T(j+1,i+1)+T(j,i)
            enddo
            f(totnodes)=-1.*(1.+2.*lam(2))*T(j+1,totnodes)+2.*lam(2)
     +*T(j+1,totnodes-1)+T(j,totnodes)
c
c send information to the tridag solver
c
            call tridag(a,b,c,f,delT,totnodes)
c
c update temperature values
c
            do i=1,totnodes
               T(j+1,i)=T(j+1,i)+delT(i)
            enddo
            b(1)=1.+2.*lam(1)+(8.*lam(1)*dx(1)*epsi*sigma*T(j+1,1)**3)
     1/k(1)
            normf=twonorm(f,totnodes)
            normdelT=twonorm(delT,totnodes)
         enddo
      time(j+1)=time(j)+dt
      enddo
      do i=1,n
         qrad(i)=sigma*epsi*T(i,1)**4
         qcond(i)=qcv(i)-qrad(i)
      enddo
      return
      end
```

```
c program that numerically predicts the heat transfer into a stack of TPS
c materials.
c
c Kris Cowart 8 Oct 99
c
      subroutine threestack(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n
     1,time,qcv,qcond,qrad)

      implicit none
      integer i,j,n,nodes(6),sum1,sum2,sum3,totnodes,matnum,count
      real*8 dt,k(6),den(6),cp(6),initT,L(6),dx(6),sigma,epsi,eps
     1,alpha(6),lam(6),a(100),b(100),c(100),T(5000,100),normf
     1,normdelT,f(100),delT(100),twonorm,Tradeq(5000),time(5000)
     1,qcond(5000),qrad(5000),qcv(5000),vecinfnorm

      external twonorm,vecinfnorm

      call linint(n,dt,time,qcv,Tradeq)
c
c define the constants
c
      eps=1.e-4
      sigma=5.67061e-8
      do i=1,matnum
         dx(i)=L(i)/(nodes(i)-1.)
         alpha(i)=k(i)/(den(i)*cp(i)*1000.)
         lam(i)=alpha(i)*dt/(dx(i)**2)
      enddo
      sum1=nodes(1)+nodes(2)
      sum2=sum1+nodes(3)
      totnodes=nodes(1)+nodes(2)+nodes(3)
c
c apply the initial conditions
c
      do i=1,totnodes
         T(1,i)=initT
      enddo
c
c make the initial guess for the temperature at the next time step
c
      do i=1,totnodes
         T(2,i)=1000.
      enddo
c
c set up the Jacobian Matrix for the system of equations
c
      a(1)=-2.*lam(1)
      b(1)=1.+2.*lam(1)+8.*lam(1)*dx(1)*epsi*sigma*(T(2,1)**3)/k(1)
      c(1)=0.
      do i=2,nodes(1)-1
         a(i)=-1.*lam(1)
         b(i)=1.+2.*lam(1)
         c(i)=-1.*lam(1)
      enddo
      a(nodes(1))=-2.*lam(2)
      b(nodes(1))=1.+2.*lam(2)+2.*lam(2)*k(1)/k(2)
      c(nodes(1))=-2.*lam(2)*k(1)/k(2)
      do i=nodes(1)+1,sum1-1
         a(i)=-1.*lam(2)
         b(i)=1.+2.*lam(2)
```

```
            c(i)=-1.*lam(2)
         enddo
         a(sum1)=-2.*lam(3)
         b(sum1)=1.+2.*lam(3)+2.*lam(3)*k(2)/k(3)
         c(sum1)=-2.*lam(3)*k(2)/k(3)
         do i=sum1+1,sum2-1
            a(i)=-1.*lam(3)
            b(i)=1.+2.*lam(3)
            c(i)=-1.*lam(3)
         enddo
         a(totnodes)=0.
         b(totnodes)=1.+2.*lam(3)
         c(totnodes)=-2.*lam(3)
c
c solution at the first time step
c
         normf=1.
         normdelT=1.
         time(1)=0.
           count=0
         dowhile(((normf.gt.eps).or.(normdelT.gt.eps)).and.(count.le.1000))
              count=count+1
            f(1)=2.*lam(1)*T(2,2)-T(2,1)-2.*lam(1)*T(2,1)-(2.*lam(1)*dx(1)
     1*epsi*sigma*T(2,1)**4)/k(1)+2.*lam(1)*dx(1)*qcv(1)/k(1)+T(1,1)
            do i=2,nodes(1)-1
               f(i)=lam(1)*T(2,i-1)-(1.+2.*lam(1))*T(2,i)+lam(1)*T(2,i+1)
     1+T(1,i)
            enddo
            f(nodes(1))=2.*lam(2)*T(2,nodes(1)+1)-(1.+2.*lam(2)+2.*lam(2)
     1*k(1)/k(2))*T(2,nodes(1))+2.*lam(2)*k(1)/k(2)*T(2,nodes(1)-1)
     1+T(1,nodes(1))
            do i=nodes(1)+1,sum1-1
               f(i)=lam(2)*T(2,i-1)-(1.+2.*lam(2))*T(2,i)+lam(2)*T(2,i+1)
     1+T(1,i)
            enddo
            f(sum1)=2.*lam(3)*T(2,sum1+1)-(1.+2.*lam(3)+2.*lam(3)*k(2)/k(3))
     1*T(2,sum1)+2.*lam(3)*k(2)/k(3)*T(2,sum1-1)+T(1,sum1)
            do i=sum1+1,sum2-1
               f(i)=lam(3)*T(2,i-1)-(1.+2.*lam(3))*T(2,i)+lam(3)*T(2,i+1)
     1+T(1,i)
            enddo
            f(totnodes)=-1.*(1.+2.*lam(3))*T(2,totnodes)+2.*lam(3)
     1*T(2,totnodes-1)+T(1,totnodes)
c
c send to tridiagonal solver
c
            call tridag(a,b,c,f,delT,totnodes)
c
c update the temperature value for this time step
c
            do i=1,totnodes
               T(2,i)=T(2,i)+delT(i)
            enddo
            b(1)=1.+2.*lam(1)+8.*lam(1)*dx(1)*epsi*sigma*(T(2,1)**3)/k(1)
            normf=twonorm(f,totnodes)
            normdelT=twonorm(delT,totnodes)
c           normf=vecinfnorm(f,totnodes)
c           normdelT=vecinfnorm(delT,totnodes)
c              write(*,*) normf,normdelT,count
          enddo
```

```
c      pause
       time(2)=time(1)+dt
c
c solution of system by marching in time
c
       do j=2,n
          do i=1,totnodes
             T(j+1,i)=T(j,i)
          enddo
          b(1)=1.+2.*lam(1)+8.*lam(1)*dx(1)*epsi*sigma*(T(j+1,1)**3)/k(1)
          normf=1.
          normdelT=1.
            count = 0
          dowhile(((normf.gt.eps).or.(normdelT.gt.eps)).and.(count.le.1000))
                count=count+1
             f(1)=2.*lam(1)*T(j+1,2)-T(j+1,1)-2.*lam(1)*T(j+1,1)
     1-(2.*lam(1)*dx(1)*epsi*sigma*T(j+1,1)**4)/k(1)+2.*lam(1)*dx(1)
     1*qcv(j)/k(1)+T(j,1)
             do i=2,nodes(1)-1
                f(i)=lam(1)*T(j+1,i-1)-(1.+2.*lam(1))*T(j+1,i)+lam(1)
     1*T(j+1,i+1)+T(j,i)
             enddo
             f(nodes(1))=2.*lam(2)*T(j+1,nodes(1)+1)-(1.+2.*lam(2)
     1+2.*lam(2)*k(1)/k(2))*T(j+1,nodes(1))+2.*lam(2)*k(1)/k(2)
     1*T(j+1,nodes(1)-1)+T(j,nodes(1))
             do i=nodes(1)+1,sum1-1
                f(i)=lam(2)*T(j+1,i-1)-(1.+2.*lam(2))*T(j+1,i)+lam(2)
     1*T(j+1,i+1)+T(j,i)
             enddo
             f(sum1)=2.*lam(3)*T(j+1,sum1+1)-(1.+2.*lam(3)+2.*lam(3)
     1*k(2)/k(3))*T(j+1,sum1)+2.*lam(3)*k(2)/k(3)*T(j+1,sum1-1)
     1+T(j,sum1)
             do i=sum1+1,sum2-1
                f(i)=lam(3)*T(j+1,i-1)-(1.+2.*lam(3))*T(j+1,i)
     1+lam(3)*T(j+1,i+1)+T(j,i)
             enddo
             f(totnodes)=-1.*(1.+2.*lam(3))*T(j+1,totnodes)+2.*lam(3)
     1*T(j+1,totnodes-1)+T(j,totnodes)
c
c send to tridiagonal solver
c
             call tridag(a,b,c,f,delT,totnodes)
c
c uptdate the temperature values
c
             do i=1,totnodes
                T(j+1,i)=T(j+1,i)+delT(i)
             enddo
             b(1)=1.+2.*lam(1)+(8.*lam(1)*dx(1)*epsi*sigma*T(j+1,1)**3)/k(1)
             normf=twonorm(f,totnodes)
             normdelT=twonorm(delT,totnodes)
c            normf=vecinfnorm(f,totnodes)
c            normdelT=vecinfnorm(delT,totnodes)
c               write(*,*) normf,normdelT,count,j
          enddo
c      pause
       time(j+1)=time(j)+dt
       enddo
       do j=1,n
          qrad(j)=epsi*sigma*T(j,1)**4
```

```
   qcond(j)=qcv(j)-qrad(j)
enddo
return
end
```

```
c program that numerically predicts the heat transfer into a stack of TPS
c materials.
c
c Kris Cowart 8 Oct 99
c
      subroutine fourstack(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n
     1,time,qcv,qcond,qrad)

      implicit none
      integer i,j,n,nodes(6),sum1,sum2,sum3,totnodes,matnum
      real*8 dt,k(6),den(6),cp(6),initT,L(6),dx(6),sigma,epsi,eps
     1,alpha(6),lam(6),a(100),b(100),c(100),T(5000,100),normf
     1,normdelT,f(100),delT(100),twonorm,Tradeq(5000),time(5000)
     1,qcond(5000),qrad(5000),qcv(5000)

      external twonorm

      call linint(n,dt,time,qcv,Tradeq)
c
c define the constants
c
      eps=1.e-6
      sigma=5.67061e-8
      do i=1,matnum
         dx(i)=L(i)/(nodes(i)-1.)
         alpha(i)=k(i)/(den(i)*cp(i)*1000.)
         lam(i)=alpha(i)*dt/(dx(i)**2)
      enddo
      sum1=nodes(1)+nodes(2)
      sum2=sum1+nodes(3)
      sum3=sum2+nodes(4)
      totnodes=nodes(1)+nodes(2)+nodes(3)+nodes(4)
c
c apply the initial conditions
c
      do i=1,totnodes
         T(1,i)=initT
      enddo
c
c make the initial guess for the temperature at the next time step
c
      do i=1,totnodes
         T(2,i)=1000.
      enddo
c
c set up the Jacobian Matrix for the system of equations
c

      a(1)=-2.*lam(1)
      b(1)=1.+2.*lam(1)+8.*lam(1)*dx(1)*epsi*sigma*(T(2,1)**3)/k(1)
      c(1)=0.
      do i=2,nodes(1)-1
         a(i)=-1.*lam(1)
         b(i)=1.+2.*lam(1)
         c(i)=-1.*lam(1)
      enddo
      a(nodes(1))=-2.*lam(2)
      b(nodes(1))=1.+2.*lam(2)+2.*lam(2)*k(1)/k(2)
      c(nodes(1))=-2.*lam(2)*k(1)/k(2)
      do i=nodes(1)+1,sum1-1
```

```
            a(i)=-1.*lam(2)
            b(i)=1.+2.*lam(2)
            c(i)=-1.*lam(2)
         enddo
         a(sum1)=-2.*lam(3)
         b(sum1)=1.+2.*lam(3)+2.*lam(3)*k(2)/k(3)
         c(sum1)=-2.*lam(3)*k(2)/k(3)
         do i=sum1+1,sum2-1
            a(i)=-1.*lam(3)
            b(i)=1.+2.*lam(3)
            c(i)=-1.*lam(3)
         enddo
         a(sum2)=-2.*lam(4)
         b(sum2)=1.+2.*lam(4)+2.*lam(4)*k(3)/k(4)
         c(sum2)=-2.*lam(4)*k(3)/k(4)
         do i=sum2+1,sum3-1
            a(i)=-1.*lam(4)
            b(i)=1.+2.*lam(4)
            c(i)=-1.*lam(4)
         enddo
         a(sum3)=0.
         b(sum3)=1.+2.*lam(4)
         c(sum3)=-2.*lam(4)
c
c solution at the first time step
c
         normf=1.
         normdelT=1.
         time(1)=0.
         dowhile((normf.gt.eps).or.(normdelT.gt.eps))
            f(1)=2.*lam(1)*T(2,2)-T(2,1)-2.*lam(1)*T(2,1)-(2.*lam(1)*dx(1)
     1*epsi*sigma*T(2,1)**4)/k(1)+2.*lam(1)*dx(1)*qcv(1)/k(1)+T(1,1)
            do i=2,nodes(1)-1
               f(i)=lam(1)*T(2,i-1)-(1.+2.*lam(1))*T(2,i)+lam(1)*T(2,i+1)
     1+T(1,i)
            enddo
            f(nodes(1))=2.*lam(2)*T(2,nodes(1)+1)-(1.+2.*lam(2)+2.*lam(2)
     1*k(1)/k(2))*T(2,nodes(1))+2.*lam(2)*k(1)/k(2)*T(2,nodes(1)-1)
     1+T(1,nodes(1))
            do i=nodes(1)+1,sum1-1
               f(i)=lam(2)*T(2,i-1)-(1.+2.*lam(2))*T(2,i)+lam(2)*T(2,i+1)
     1+T(1,i)
            enddo
            f(sum1)=2.*lam(3)*T(2,sum1+1)-(1.+2.*lam(3)+2.*lam(3)*k(2)/k(3))*
     1T(2,sum1)+2.*lam(3)*k(2)/k(3)*T(2,sum1-1)+T(1,sum1)
            do i=sum1+1,sum2-1
               f(i)=lam(3)*T(2,i-1)-(1.+2.*lam(3))*T(2,i)+lam(3)*T(2,i+1)
     1+T(1,i)
            enddo
            f(sum2)=2.*lam(4)*T(2,sum2+1)-(1.+2.*lam(4)+2.*lam(4)*k(3)/k(4))*
     1T(2,sum2)+2.*lam(4)*k(3)/k(4)*T(2,sum2-1)+T(1,sum2)
            do i=sum2+1,sum3-1
               f(i)=lam(4)*T(2,i-1)-(1.+2.*lam(4))*T(2,i)+lam(4)*T(2,i+1)+T(1,i)
            enddo
            f(sum3)=-1.*(1.+2.*lam(4))*T(2,sum3)+2.*lam(4)*T(2,sum3-1)
     ++T(1,sum3)

c
c send to trisdiagonal solver
c
```

```
          call tridag(a,b,c,f,delT,totnodes)
c
c update the temperature values for this time step
c
          do i=1,totnodes
             T(2,i)=T(2,i)+delT(i)
          enddo
          b(1)=1.+2.*lam(1)+8.*lam(1)*dx(1)*epsi*sigma*(T(2,1)**3)/k(1)
          normf=twonorm(f,totnodes)
          normdelT=twonorm(delT,totnodes)
       enddo
       time(j)=time(j)+dt

c
c solution of system by marching in time
c
       do j=2,n
          do i=1,totnodes
             T(j+1,i)=T(j,i)
          enddo
          b(1)=1.+2.*lam(1)+8.*lam(1)*dx(1)*epsi*sigma*(T(j+1,1)**3)/k(1)
          normf=1.
          normdelT=1.
          dowhile((normf.gt.eps).or.(normdelT.gt.eps))
             f(1)=2.*lam(1)*T(j+1,2)-T(j+1,1)-2.*lam(1)*T(j+1,1)-(2.*lam(1)
     1*dx(1)*epsi*sigma*T(j+1,1)**4)/k(1)+2.*lam(1)*dx(1)*qcv(j)/k(1)
     1+T(j,1)
             do i=2,nodes(1)-1
                f(i)=lam(1)*T(j+1,i-1)-(1.+2.*lam(1))*T(j+1,i)+lam(1)
     1*T(j+1,i+1)+T(j,i)
             enddo
             f(nodes(1))=2.*lam(2)*T(j+1,nodes(1)+1)-(1.+2.*lam(2)+2.*lam(2)
     1*k(1)/k(2))*T(j+1,nodes(1))+2.*lam(2)*k(1)/k(2)*T(j+1,nodes(1)-1)
     1+T(j,nodes(1))
             do i=nodes(1)+1,sum1-1
                f(i)=lam(2)*T(j+1,i-1)-(1.+2.*lam(2))*T(j+1,i)+lam(2)
     1*T(j+1,i+1)+T(j,i)
             enddo
             f(sum1)=2.*lam(3)*T(j+1,sum1+1)-(1.+2.*lam(3)+2.*lam(3)
     1*k(2)/k(3))*T(j+1,sum1)+2.*lam(3)*k(2)/k(3)*T(j+1,sum1-1)+
     1T(j,sum1)
             do i=sum1+1,sum2-1
                f(i)=lam(3)*T(j+1,i-1)-(1.+2.*lam(3))*T(j+1,i)+lam(3)
     1*T(j+1,i+1)+T(j,i)
             enddo
             f(sum2)=2.*lam(4)*T(j+1,sum2+1)-(1.+2.*lam(4)+2.*lam(4)
     1*k(3)/k(4))*T(j+1,sum2)+2.*lam(4)*k(3)/k(4)*T(j+1,sum2-1)+
     1T(j,sum2)
             do i=sum2+1,sum3-1
                f(i)=lam(4)*T(j+1,i-1)-(1.+2.*lam(4))*T(j+1,i)+lam(4)
     1*T(j+1,i+1)+T(j,i)
             enddo
             f(sum3)=-1.*(1.+2.*lam(4))*T(j+1,sum3)+2.*lam(4)*T(j+1,sum3-1)
     1+T(j,sum3)
c
c send to tridiagonal solver
c
             call tridag(a,b,c,f,delT,totnodes)
c
c update the temperature values
```

```
c
            do i=1,totnodes
               T(j+1,i)=T(j+1,i)+delT(i)
            enddo
            b(1)=1.+2.*lam(1)+(8.*lam(1)*dx(1)*epsi*sigma*T(j+1,1)**3)/k(1)
            normf=twonorm(f,totnodes)
            normdelT=twonorm(delT,totnodes)
         enddo
      time(j)=time(j)+dt
      enddo
      do j=1,n
         qrad(j)=epsi*sigma*T(j,1)**4
         qcond(j)=qcv(j)-qrad(j)
      enddo
      return
      end
```

```
c program that numerically predicts the heat transfer into a stack of TPS
c materials.
c
c Kris Cowart 9 Jan 00
c
      subroutine fivestack(matnum,dt,nodes,epsi,k,den,cp,initT,L,T,n
     1,time,qcv,qcond,qrad)

      implicit none
      integer i,j,n,nodes(6),sum1,sum2,sum3,totnodes,matnum,count
      real*8 dt,k(6),den(6),cp(6),initT,L(6),dx(6),sigma,epsi,eps
     1,alpha(6),lam(6),a(100),b(100),c(100),T(5000,100),normf
     1,normdelT,f(100),delT(100),twonorm,Tradeq(5000),time(5000)
     1,qcond(5000),qrad(5000),qcv(5000),g(24),obj,x(12)

      external twonorm

      call linint(n,dt,time,qcv,Tradeq)
c
c define the constants
c
      eps=1.e-6
      sigma=5.67061e-8
      epsi=0.83
      do i=1,matnum
         dx(i)=L(i)/(nodes(i)-1.)
         alpha(i)=k(i)/(den(i)*cp(i)*1000.)
         lam(i)=alpha(i)*dt/(dx(i)**2)
      enddo
      sum1=nodes(1)+nodes(2)
      sum2=sum1+nodes(3)
      sum3=sum2+nodes(4)
      totnodes=nodes(1)+nodes(2)+nodes(3)+nodes(4)+nodes(5)
c
c apply the initial conditions
c
      do i=1,totnodes
         T(1,i)=initT
      enddo
c
c make the initial guess for the temperature at the next time step
c
      do i=1,totnodes
         T(2,i)=1000.
      enddo
c
c set up the Jacobian Matrix for the system of equations
c
      a(1)=-2.*lam(1)
      b(1)=1.+2.*lam(1)+8.*lam(1)*dx(1)*epsi*sigma*(T(2,1)**3)/k(1)
      c(1)=0.
      do i=2,nodes(1)-1
         a(i)=-1.*lam(1)
         b(i)=1.+2.*lam(1)
         c(i)=-1.*lam(1)
      enddo
      a(nodes(1))=-2.*lam(2)
      b(nodes(1))=1.+2.*lam(2)+2.*lam(2)*k(1)/k(2)
      c(nodes(1))=-2.*lam(2)*k(1)/k(2)
      do i=nodes(1)+1,sum1-1
```

```
            a(i)=-1.*lam(2)
            b(i)=1.+2.*lam(2)
            c(i)=-1.*lam(2)
         enddo
         a(sum1)=-2.*lam(3)
         b(sum1)=1.+2.*lam(3)+2.*lam(3)*k(2)/k(3)
         c(sum1)=-2.*lam(3)*k(2)/k(3)
         do i=sum1+1,sum2-1
            a(i)=-1.*lam(3)
            b(i)=1.+2.*lam(3)
            c(i)=-1.*lam(3)
         enddo
         a(sum2)=-2.*lam(4)
         b(sum2)=1.+2.*lam(4)+2.*lam(4)*k(3)/k(4)
         c(sum2)=-2.*lam(4)*k(3)/k(4)
         do i=sum2+1,sum3-1
            a(i)=-1.*lam(4)
            b(i)=1.+2.*lam(4)
            c(i)=-1.*lam(4)
         enddo
         a(sum3)=-2.*lam(5)
         b(sum3)=1.+2.*lam(5)+2.*lam(5)*k(4)/k(5)
         c(sum3)=-2.*lam(5)*k(4)/k(5)
         do i=sum3+1,totnodes-1
            a(i)=-1.*lam(5)
            b(i)=1.+2.*lam(5)
            c(i)=-1.*lam(5)
         enddo
         a(totnodes)=0.
         b(totnodes)=1.+2.*lam(5)
         c(totnodes)=-2.*lam(5)
c
c solution at the first time step
c
         normf=1.
         normdelT=1.
         time(1)=0.
           count=0
         dowhile(((normf.gt.eps).or.(normdelT.gt.eps)).and.(count.le.1000))
              count=count+1
            f(1)=2.*lam(1)*T(2,2)-T(2,1)-2.*lam(1)*T(2,1)-
       1(2.*lam(1)*dx(1)*epsi*sigma*T(2,1)**4)/k(1)+2.
       1*lam(1)*dx(1)*qcv(1)/k(1)+T(1,1)
            do i=2,nodes(1)-1
               f(i)=lam(1)*T(2,i-1)-(1.+2.*lam(1))*T(2,i)
       1+lam(1)*T(2,i+1)+T(1,i)
            enddo
            f(nodes(1))=2.*lam(2)*T(2,nodes(1)+1)-(1.+2.*lam(2)
       1+2.*lam(2)*k(1)/k(2))*T(2,nodes(1))+2.*lam(2)*k(1)/k(2)
       1*T(2,nodes(1)-1)+T(1,nodes(1))
            do i=nodes(1)+1,sum1-1
               f(i)=lam(2)*T(2,i-1)-(1.+2.*lam(2))*T(2,i)+lam(2)*T(2,i+1)+T(1,i)
            enddo
            f(sum1)=2.*lam(3)*T(2,sum1+1)-(1.+2.*lam(3)+2.*lam(3)*k(2)/k(3))*
       1T(2,sum1)+2.*lam(3)*k(2)/k(3)*T(2,sum1-1)+T(1,sum1)
            do i=sum1+1,sum2-1
               f(i)=lam(3)*T(2,i-1)-(1.+2.*lam(3))*T(2,i)+lam(3)*T(2,i+1)+T(1,i)
            enddo
            f(sum2)=2.*lam(4)*T(2,sum2+1)-(1.+2.*lam(4)+2.*lam(4)*k(3)/k(4))*
       1T(2,sum2)+2.*lam(4)*k(3)/k(4)*T(2,sum2-1)+T(1,sum2)
```

```
          do i=sum2+1,sum3-1
             f(i)=lam(4)*T(2,i-1)-(1.+2.*lam(4))*T(2,i)+lam(4)*T(2,i+1)+T(1,i)
          enddo
          f(sum3)=2.*lam(5)*T(2,sum3+1)-(1.+2.*lam(5)+2.*lam(5)*k(4)/k(5))*
     +T(2,sum3)+2.*lam(5)*k(4)/k(5)*T(2,sum3-1)+T(1,sum3)
          do i=sum3+1,totnodes-1
             f(i)=lam(5)*T(2,i-1)-(1.+2.*lam(5))*T(2,i)+lam(5)*T(2,i+1)+T(1,i)
          enddo
          f(totnodes)=-1.*(1.+2.*lam(5))*T(2,totnodes)+2.*lam(5)*T(2,totnodes-1)
     ++T(1,totnodes)
c
c send to trisdiagonal solver
c
          call tridag(a,b,c,f,delT,totnodes)
c
c update the temperature values for this time step
c
          do i=1,totnodes
             T(2,i)=T(2,i)+delT(i)
          enddo
          b(1)=1.+2.*lam(1)+8.*lam(1)*dx(1)*epsi*sigma*(T(2,1)**3)/k(1)
          normf=twonorm(f,totnodes)
          normdelT=twonorm(delT,totnodes)
c            write(*,*) normf,normdelT
       enddo
       time(j)=time(j)+dt
c
c solution of system by marching in time
c
       do j=2,n
          do i=1,totnodes
             T(j+1,i)=T(j,i)
          enddo
          b(1)=1.+2.*lam(1)+8.*lam(1)*dx(1)*epsi*sigma*(T(j+1,1)**3)/k(1)
          normf=1.
          normdelT=1.
            count=0
          dowhile(((normf.gt.eps).or.(normdelT.gt.eps)).and.(count.le.1000))
               count=count+1
             f(1)=2.*lam(1)*T(j+1,2)-T(j+1,1)-2.*lam(1)*T(j+1,1)-
     1(2.*lam(1)*dx(1)*epsi*sigma*T(j+1,1)**4)/k(1)
     1+2.*lam(1)*dx(1)*qcv(j)/k(1)+T(j,1)
             do i=2,nodes(1)-1
                f(i)=lam(1)*T(j+1,i-1)-(1.+2.*lam(1))*T(j+1,i)
     1+lam(1)*T(j+1,i+1)+T(j,i)
             enddo
      f(nodes(1))=2.*lam(2)*T(j+1,nodes(1)+1)-(1.+2.*lam(2)
     1+2.*lam(2)*k(1)/k(2))*T(j+1,nodes(1))+2.*lam(2)*k(1)/k(2)
     1*T(j+1,nodes(1)-1)+T(j,nodes(1))
             do i=nodes(1)+1,sum1-1
                f(i)=lam(2)*T(j+1,i-1)-(1.+2.*lam(2))*T(j+1,i)+lam(2)
     1*T(j+1,i+1)+T(j,i)
             enddo
             f(sum1)=2.*lam(3)*T(j+1,sum1+1)-(1.+2.*lam(3)+2.*lam(3)
     1*k(2)/k(3))*T(j+1,sum1)+2.*lam(3)*k(2)/k(3)*T(j+1,sum1-1)
     1+T(j,sum1)
             do i=sum1+1,sum2-1
                f(i)=lam(3)*T(j+1,i-1)-(1.+2.*lam(3))*T(j+1,i)+lam(3)
     1*T(j+1,i+1)+T(j,i)
             enddo
```

```
              f(sum2)=2.*lam(4)*T(j+1,sum2+1)-(1.+2.*lam(4)+2.*lam(4)
     1*k(3)/k(4))*T(j+1,sum2)+2.*lam(4)*k(3)/k(4)*T(j+1,sum2-1)
     1+T(j,sum2)
              do i=sum2+1,sum3-1
                 f(i)=lam(4)*T(j+1,i-1)-(1.+2.*lam(4))*T(j+1,i)+lam(4)
     1*T(j+1,i+1)+T(j,i)
              enddo
              f(sum3)=2.*lam(5)*T(j+1,sum3+1)-(1.+2.*lam(5)+2.*lam(5)
     1*k(4)/k(5))*T(j+1,sum3)+2.*lam(5)*k(4)/k(5)*T(j+1,sum3-1)
     1+T(j,sum3)
              do i=sum3+1,totnodes-1
                 f(i)=lam(5)*T(j+1,i-1)-(1.+2.*lam(5))*T(j+1,i)+lam(5)
     1*T(j+1,i+1)+T(j,i)
              enddo
              f(totnodes)=-1.*(1.+2.*lam(5))*T(j+1,totnodes)+2.*lam(5)
     1*T(j+1,totnodes-1)+T(j,totnodes)
c
c send to tridiagonal solver
c
              call tridag(a,b,c,f,delT,totnodes)
c
c update the temperature values
c
              do i=1,totnodes
                 T(j+1,i)=T(j+1,i)+delT(i)
              enddo
              b(1)=1.+2.*lam(1)+(8.*lam(1)*dx(1)*epsi*sigma*T(j+1,1)**3)/k(1)
              normf=twonorm(f,totnodes)
              normdelT=twonorm(delT,totnodes)
c                write(*,*) normf,normdelT
           enddo
        time(j)=time(j)+dt
        enddo
        do j=1,n
           qrad(j)=epsi*sigma*T(j,1)**4
           qcond(j)=qcv(j)-qrad(j)
        enddo
        return
        end
```

```
      subroutine linint(n,dt,time,qcon,Tradeq)
c
c program to linearly interpolate input values for TCAT
c
      implicit none
      integer i,j,numdata,n,start
      real*8 time(5000),Tradeq(5000),qcon(5000)
      real*8 tidata(5000),Tdata(5000),qdata(5000),dt

      open(50,file='miniver.in')
      read(50,*)numdata
      do i=1,numdata
         read(50,*)tidata(i)
         read(50,*)qdata(i)
         read(50,*)Tdata(i)
      enddo
      close(50)
      start=tidata(1)
      n=(tidata(numdata)-start)/dt
      time(1)=tidata(1)
      do i=1,n+1
         do j=1,numdata
            if(time(i).eq.tidata(j))then
               Tradeq(i)=Tdata(j)
               qcon(i)=qdata(j)
            endif
            if((time(i).gt.tidata(j)).and.(time(i).lt.tidata(j+1)))
     +then
              Tradeq(i)=(time(i)-tidata(j))/(tidata(j+1)-tidata(j))*
     +(Tdata(j+1)-Tdata(j))+Tdata(j)
              qcon(i)=(time(i)-tidata(j))/(tidata(j+1)-tidata(j))*
     +(qdata(j+1)-qdata(j))+qdata(j)
            endif
         enddo
         time(i+1)=time(i)+dt
      enddo
      open(61,file='linear.out')
      do i=1,n+1
         write(61,1)time(i),Tradeq(i),qcon(i)
      enddo
 1    format(f12.2,f20.2,f20.6)
 2    format(i2)
 3    format(f10.2,f10.2,f10.2)
      close(61)
      return
      end
```

```
      subroutine tridag(a,b,c,r,u,n)
*
* Solves for a vector u(1:n) of length N the tridiagonal linear set given.
* a(1:n),b(1:n),c(1:n), and r(1:n) are input vectors and are not modified.
* Parameter nmax is the maximum expected velue of N.
*
      implicit none
      integer n,nmax
      real*8 a(n),b(n),c(n),r(n),u(n)
      parameter (nmax=500)
      integer j
      real*8 bet,gam(nmax)

      if(b(1).eq.0.) pause 'tridag:rewrite equations'
      bet=b(1)
      u(1)=r(1)/bet
      do j=2,n
         gam(j)=c(j-1)/bet
         bet=b(j)-a(j)*gam(j)
         if(bet.eq.0.)pause 'tridag failed'
         u(j)=(r(j)-a(j)*u(j-1))/bet
      enddo
      do j=n-1,1,-1
         u(j)=u(j)-gam(j+1)*u(j+1)
      enddo
      return
      end



      function twonorm(x,n)
c
c external function to calculate the twonorm of a vector
c
      implicit none
      integer i,n
      real*8 x(n),twonorm,sum

      do i=1,n
         x(i)=x(i)**2
      enddo
      sum=x(1)
      do i=2,n
         sum=sum+x(i)
      enddo
      twonorm=sqrt(sum)
      return
      end
```

APPENDIX C

TCAT SINGLE TPS DESIGN CGI SCRIPTS

```
#!/usr/sbin/perl

# Author: Kris Cowart
# Date Created: 8 Feb 00
#----------------------- Obtain Information From Web Browser------------------
-----------
#
# This section of code allows a cgi script to run from a web browser.
# The information is passed using the POST and the input information is
# parceled out in a mesh type variable and stores in $FORM{$name}.
#

read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
       ($name, $value) = split(/=/, $pair);
       $value =~ tr/+/ /;
       $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
       $value =~ s/~!/ ~!/g;
       $FORM{$name} = $value;
}

#-------- assign information from input website to appropriate variable -------

$bodyarea=$FORM{'bodyarea'};
$bodytile=$FORM{'bodytile'};
$bodyblanket=$FORM{'bodyblanket'};
$sharp=$FORM{'sharp'};
$filename=$FORM{'filename'};
$bodyside=$FORM{'side'};
$material=$FORM{'tps'};

#----------------- MINIVER file is partitioned into body point files ----------
-----------
#
# partition the MINIVER file into separate body point files that includes time,
heat rate, temp
#
$i=0;                           #initialization of counter index "i"
$filecount=0;
open(FILE,"<$filename");        #open statement with filehandle FILE

until(eof FILE) {               #goes through the MINVER file until the end is
reached
   START: $_=<FILE>;            #skips the first line of the data file
   if (/^.{73}(\S*)/) {                #searches for the bodypoint label
   $title=$1;

   system("touch point$title");        #create body point file
   system("chmod 777 point$title");    #change permissions inorder to write to
body point file

   $filecount=$filecount+1;
   } # end of if statement

   while(<FILE>) {                     #while loop that reads thru each line of
the file
       $i=$i+1;
```

```
/^\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S
*)\s*(\S*)\s*(\S*)\s*(\S*)/;
      $time[$i]=$1;              #array declaration for time
      $tradeq[$i]=$9;                  #array declaration for rad eq temp
      $qconv[$i]=$10;                  #array declaration for conv heat rate
      $n=$i;
      if ($1==-100) {
         open(NEWFILE,">point$title"); #open statement for new file "newsample"
        $timesteps=$n-1;
         printf NEWFILE "$timesteps \n";#formatted print statement"
        for($i=1; $i<$n; $i++) {
            printf NEWFILE "%f \n",$time[$i];
            printf NEWFILE "%f \n",$qconv[$i];
            printf NEWFILE "%f \n",$tradeq[$i];
         }  #end of for loop

         close(NEWFILE);
         $i=0;
         goto START
      }  #end of if statement

   }  #end of while loop
}  #end of until loop
close(FILE);
#--------------------------
# incrementally move each body point file into "miniver.in" and execute tcat
# perl script that creates the filenames needed inorder to batch execute tcat
# filename are created via input from the user
# inputs (file_start_number, file_end_number, body_side)

$start=1;             #start of the file index
$end=($filecount-1)/2;    #end of the file index
#$end=1;

open(FILE,">inputs_for_outputs");
printf FILE "$bodyside\n";
printf FILE "$filename\n";
printf FILE "$end\n";
close(FILE);

system("touch thickness.txt");
system("chmod 777 thickness.txt");
open(FINALOUTPUT,">thickness.txt");
print FINALOUTPUT "TPS design results for $bodyside analysis for $filename
\n\n";
#-------------
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {

   if ($bodyside eq "leeward") {
      $name="pointl$loop_index";
   }
   if ($bodyside eq "windward") {
      $name="pointw$loop_index";
   }
   system("touch /home/asdl1/kcowart/public_html/miniver.in");
   system("chmod 777 /home/asdl1/kcowart/public_html/miniver.in");
   `cp /home/asdl1/kcowart/public_html/$name
/home/asdl1/kcowart/public_html/miniver.in`;
```

```
# ---------------- assign material properties -------------------------------
   if ($material eq "AETB8_Al") {
      $inputfile="AETB_8_AL_5inputs.in";
   }
   if ($material eq "AETB8_GrEx") {
      $inputfile="AETB_8_GrEx_5inputs.in";
   }
   if ($material eq "AETB12_Al") {
      $inputfile="AETB_12_AL_5inputs.in";
   }
   if ($material eq "AETB12_GrEx") {
      $inputfile="AETB_12_GrEX_5inputs.in";
   }
   if ($material eq "FRCI12_Al") {
      $inputfile="FRCI_12_AL_5inputs.i";
   }
   if ($material eq "FRCI12_GrEx") {
      $inputfile="FRCI_12_GrEx_5inputs.in";
   }
   if ($material eq "FRCI20_Al") {
      $inputfile="FRCI_20_AL_5inputs.in";
   }
   if ($material eq "FRCI20_GrEx") {
      $inputfile="FRCI_20_GrEx_5inputs.in";
   }
   if ($material eq "LI900_Al") {
      $inputfile="LI_900_AL_5inputs.in";
   }
   if ($material eq "LI900_GrEx") {
      $inputfile="LI_900_GrEx_5inputs.in";
   }
   if ($material eq "LI2200_Al") {
      $inputfile="LI_2200_AL_5inputs.in";
   }
   if ($material eq "LI2200_GrEx") {
      $inputfile="LI_2200_GrEx_5inputs.in";
   }
   if ($material eq "RCC_GrEx") {
      $inputfile="RCC_GrEx_5inputs.in";
   }
   if ($material eq "RCC_Al") {
      $inputfile="RCC_AL_5inputs.in";
   }
   if ($material eq "SiC_GrEx") {
      $inputfile="SIC_GrEx_5inputs.in";
   }
   if ($material eq "SiC_Al") {
      $inputfile="SIC_AL_5inputs.in";
   }
   if ($material eq "TUFI_GrEx") {
      $inputfile="TUFI_GrEx_5inputs.in";
   }
   if ($material eq "AFRSI_Al") {
      $inputfile="AFRSI_Al_3inputs.in";
   }
   if ($material eq "AFRSI_GrEx") {
      $inputfile="AFRSI_GrEx_3inputs.in";
   }
      if ($material eq "CFBI_Al") {
      $inputfile="CFBI_Al_3inputs.in";
```

```
   }
      if ($material eq "CFBI_GrEx") {
      $inputfile="CFBI_GrEx_3inputs.in";
   }
      if ($material eq "DURAFRSI_Al") {
      $inputfile="DURAFRSI_Al_3inputs.in";
   }
      if ($material eq "DURAFRSI_GrEx") {
      $inputfile="DURAFRSI_GrEx_3inputs.in";
   }
      if ($material eq "PBI_Al") {
      $inputfile="PBI_Al_3inputs.in";
   }
      if ($material eq "PBI_GrEx") {
      $inputfile="PBI_GrEx_3inputs.in";
   }
      if ($material eq "TABI_Al") {
      $inputfile="TABI_Al_3inputs.in";
   }
      if ($material eq "TABI_GrEx") {
      $inputfile="TABI_GrEx_3inputs.in";
   }
# -------------

   system("touch inputs.in");
   system("chmod 777 inputs.in");
   system("cp $inputfile inputs.in >> error 2>&1");

   if ($material eq "RCC_GrEx") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material eq "RCC_Al") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material eq "SiC_GrEx") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material eq "SiC_Al") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material eq "TUFI_GrEx") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   else {
      system("/home/asdl1/kcowart/public_html/go.exe>junkoutput");# run TCAT
   }

#   system("touch $name[$loop_index]_temphist");
#   system("chmod 777 $name[$loop_index]_temphist");
   system("mv fort.90 ${name}_temphist >> error3");


#   system("touch $name[$loop_index]_heatratehist");
#   system("chmod 777 $name[$loop_index]_heatratehist");
   system("mv fort.91 ${name}_heatratehist >> error4");
```

```
   open(TCATOUTPUT,"<fort.12"); # obtain the current thickness of TPS file
   $_=<TCATOUTPUT>;
   /^\s*(\S*)/;
   $thickness=$1;
   close(TCATOUTPUT);

   $thkns[$loop_index]=$thickness;

   printf FINALOUTPUT "%8s - TPS thickness = %6.3f
in.\n",$name,$thickness*100/2.54;
}

#system("rm point*");

# ---------- end of heating analysis ----------------
$avg_thickness=0.0;
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   $avg_thickness=$avg_thickness+$thkns[$loop_index];
}
$avg_thickness=$avg_thickness/$end;

open(DEN,"<material_density"); # obtain the density of TPS material
$_=<DEN>;
/^\s*(\S*)/;
$tps_density=$1;
close(DEN);

if (($sharp eq "yes") && ($bodyside eq "windward")) {
   $sharp_weight=4.6e-3*$bodyarea;      #SHARP weight in lbm
   if ($material eq "RCC_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1577.8347)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "RCC_Al") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1577.8347)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "SiC_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*2400.00)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "SiC_Al") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*2400.00)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "TUFI_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1313.53)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
```

```
   else {
$tile_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2808
;      #obtain density of tps
   }
   $tile_area_to_body_area=$bodytile;
   printf FINALOUTPUT "\n";
   printf FINALOUTPUT "Chosen TPS material was $material\n\n";
   printf FINALOUTPUT "Nose SHARP TPS: %6.2f lbm\n", $sharp_weight;
   printf FINALOUTPUT "$material unit weight: %8.4f lbm/ft2 \n
",$tile_unit_weight;
   printf FINALOUTPUT "$material TPS area to body area ratio: %4.2f \n
",$tile_area_to_body_area;

}
if (($sharp eq "yes") && ($bodyside eq "leeward")) {
   $sharp_weight=4.6e-3*$bodyarea;     #SHARP weight in lbm

$blanket_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2
808;; #obtain density of tps
   $blanket_area_to_body_area=$bodyblanket;
   printf FINALOUTPUT "\n";
   printf FINALOUTPUT "Chosen TPS material was $material\n\n";
   printf FINALOUTPUT "Nose SHARP TPS: %6.2f lbm\n", $sharp_weight;
   printf FINALOUTPUT "$material unit weight: %6.2f lbm/ft2 \n
",$blanket_unit_weight;
   printf FINALOUTPUT "$material TPS area to body area ratio: %6.2f \n
",$blanket_area_to_body_area;
}
if (($sharp eq "no") && ($bodyside eq "windward")) {
   if ($material eq "RCC_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1577.8347)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "RCC_Al") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1577.8347)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "SiC_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*2400.00)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "SiC_Al") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*2400.00)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "TUFI_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1313.53)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
```

```
   else {
$tile_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2808
;      #obtain density of tps
   }
   $tile_area_to_body_area=$bodytile;
   printf FINALOUTPUT "Chosen TPS material was $material\n\n";
   printf FINALOUTPUT "$material unit weight: %6.2f lbm/ft2 \n
",$tile_unit_weight;
   printf FINALOUTPUT "$material TPS area to body area ratio: %6.2f \n
",$tile_area_to_body_area;
}
if (($sharp eq "no") && ($bodyside eq "leeward")) {
$blanket_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2
808;; #obtain density of tps
   $blanket_area_to_body_area=$bodyblanket;
   printf FINALOUTPUT "Chosen TPS material was $material\n\n";
   printf FINALOUTPUT "$material unit weight: %6.2f lbm/ft2 \n
",$blanket_unit_weight;
   printf FINALOUTPUT "$material TPS area to body area ratio: %6.2f \n
",$blanket_area_to_body_area;
}
close(FINALOUTPUT);

#---------------------- print needed information to website ------------------
------------
#
# print information to webpage "tcat_output.html"
#
open(OUT,"<thickness.txt");           #opens the input file for TCAT script
open(WEB,"<tcat_output.html");   #open tcat html file
print "Content-type:text/html\n\n";
while(<WEB>){
   if(/Insert stuff here/){
      print "<BR>\n";
      while(<OUT>){
         print "<BR>\n";
        print $_;
      }
   }
   else{
      print $_;
   }
}
close(WEB);
close(OUT);
```

```
#!/usr/sbin/perl

# Author: Kris Cowart
# Date Created: 8 Feb 00
#------------------------ Obtain Information From Web Browser------------------
-----------
#
# This section of code allows a cgi script to run from a web browser.
# The information is passed using the POST and the input information is
# parceled out in a mesh type variable and stores in $FORM{$name}.
#

read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
       ($name, $value) = split(/=/, $pair);
       $value =~ tr/+/ /;
       $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
       $value =~ s/~!/ ~!/g;
       $FORM{$name} = $value;
}

#-------- assign information from input website to appropriate variable -------

$cowlarea=$FORM{'cowlarea'};
$cowltile=$FORM{'cowltile'};
$cowlblanket=$FORM{'cowlblanket'};
$sharp=$FORM{'sharp'};
$cowl_length=$FORM{'leadingedge'};
$filename=$FORM{'filename'};
$bodyside=$FORM{'side'};
$material=$FORM{'tps'};

#------------------ MINIVER file is partitioned into body point files ---------
-----------
#
# partition the MINIVER file into separate body point files that includes time,
heat rate, temp
#
$i=0;                            #initialization of counter index "i"
$filecount=0;
open(FILE,"<$filename");         #open statement with filehandle FILE

until(eof FILE) {                #goes through the MINVER file until the end is
reached
   START: $_=<FILE>;             #skips the first line of the data file
   if (/^.{73}(\S*)/) {                #searches for the bodypoint label
   $title=$1;

   system("touch point$title");        #create body point file
   system("chmod 777 point$title");    #change permissions inorder to write to
body point file

   $filecount=$filecount+1;
   } # end of if statement

   while(<FILE>) {                     #while loop that reads thru each line of
the file
      $i=$i+1;
```

```
/^\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S
*)\s*(\S*)\s*(\S*)\s*(\S*)/;
      $time[$i]=$1;                 #array declaration for time
      $tradeq[$i]=$9;                      #array declaration for rad eq temp
      $qconv[$i]=$10;                      #array declaration for conv heat rate
      $n=$i;
      if ($1==-100) {
         open(NEWFILE,">point$title"); #open statement for new file "newsample"
        $timesteps=$n-1;
         printf NEWFILE "$timesteps \n";#formatted print statement"
        for($i=1; $i<$n; $i++) {
            printf NEWFILE "%f \n",$time[$i];
            printf NEWFILE "%f \n",$qconv[$i];
            printf NEWFILE "%f \n",$tradeq[$i];
         }  #end of for loop

         close(NEWFILE);
         $i=0;
         goto START
      }  #end of if statement

   }  #end of while loop
}  #end of until loop
close(FILE);
#---------------------------
# incrementally move each body point file into "miniver.in" and execute tcat
# perl script that creates the filenames needed inorder to batch execute tcat
# filename are created via input from the user
# inputs (file_start_number, file_end_number, body_side)

$start=1;             #start of the file index
$end=($filecount-1)/2;    #end of the file index
#$end=1;

open(FILE,">inputs_for_outputs");
printf FILE "$bodyside\n";
printf FILE "$filename\n";
printf FILE "$end\n";
close(FILE);

system("touch thickness.txt");
system("chmod 777 thickness.txt");
open(FINALOUTPUT,">thickness.txt");
print FINALOUTPUT "TPS design results for $bodyside analysis for $filename
\n\n";
#------------
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {

   if ($bodyside eq "leeward") {
      $name="pointl$loop_index";
   }
   if ($bodyside eq "windward") {
      $name="pointw$loop_index";
   }
   system("touch /home/asdl1/kcowart/public_html/miniver.in");
   system("chmod 777 /home/asdl1/kcowart/public_html/miniver.in");
   `cp /home/asdl1/kcowart/public_html/$name
/home/asdl1/kcowart/public_html/miniver.in`;
```

```
# --------------- assign material properties -----------------------------
  if ($material eq "AETB8_Al") {
     $inputfile="AETB_8_AL_5inputs.in";
  }
  if ($material eq "AETB8_GrEx") {
     $inputfile="AETB_8_GrEx_5inputs.in";
  }
  if ($material eq "AETB12_Al") {
     $inputfile="AETB_12_AL_5inputs.in";
  }
  if ($material eq "AETB12_GrEx") {
     $inputfile="AETB_12_GrEx_5inputs.in";
  }
  if ($material eq "FRCI12_Al") {
     $inputfile="FRCI_12_AL_5inputs.i";
  }
  if ($material eq "FRCI12_GrEx") {
     $inputfile="FRCI_12_GrEx_5inputs.in";
  }
  if ($material eq "FRCI20_Al") {
     $inputfile="FRCI_20_AL_5inputs.in";
  }
  if ($material eq "FRCI20_GrEx") {
     $inputfile="FRCI_20_GrEx_5inputs.in";
  }
  if ($material eq "LI900_Al") {
     $inputfile="LI_900_AL_5inputs.in";
  }
  if ($material eq "LI900_GrEx") {
     $inputfile="LI_900_GrEx_5inputs.in";
  }
  if ($material eq "LI2200_Al") {
     $inputfile="LI_2200_AL_5inputs.in";
  }
  if ($material eq "LI2200_GrEx") {
     $inputfile="LI_2200_GrEx_5inputs.in";
  }
  if ($material eq "RCC_GrEx") {
     $inputfile="RCC_GrEx_5inputs.in";
  }
  if ($material eq "RCC_Al") {
     $inputfile="RCC_AL_5inputs.in";
  }
  if ($material eq "SiC_GrEx") {
     $inputfile="SIC_GrEx_5inputs.in";
  }
  if ($material eq "SiC_Al") {
     $inputfile="SIC_AL_5inputs.in";
  }
  if ($material eq "TUFI_GrEx") {
     $inputfile="TUFI_GrEx_5inputs.in";
  }
  if ($material eq "AFRSI_Al") {
     $inputfile="AFRSI_Al_3inputs.in";
  }
  if ($material eq "AFRSI_GrEx") {
     $inputfile="AFRSI_GrEx_3inputs.in";
  }
     if ($material eq "CFBI_Al") {
     $inputfile="CFBI_Al_3inputs.in";
```

```
   }
      if ($material eq "CFBI_GrEx") {
      $inputfile="CFBI_GrEx_3inputs.in";
   }
      if ($material eq "DURAFRSI_Al") {
      $inputfile="DURAFRSI_Al_3inputs.in";
   }
      if ($material eq "DURAFRSI_GrEx") {
      $inputfile="DURAFRSI_GrEx_3inputs.in";
   }
      if ($material eq "PBI_Al") {
      $inputfile="PBI_Al_3inputs.in";
   }
      if ($material eq "PBI_GrEx") {
      $inputfile="PBI_GrEx_3inputs.in";
   }
      if ($material eq "TABI_Al") {
      $inputfile="TABI_Al_3inputs.in";
   }
      if ($material eq "TABI_GrEx") {
      $inputfile="TABI_GrEx_3inputs.in";
   }
# -------------
   system("touch inputs.in");
   system("chmod 777 inputs.in");
   system("cp $inputfile inputs.in >> error 2>&1");

   if ($material eq "RCC_GrEx") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material eq "RCC_Al") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material eq "SiC_GrEx") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material eq "SiC_Al") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material eq "TUFI_GrEx") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   else {
      system("/home/asdl1/kcowart/public_html/go.exe>junkoutput");# run TCAT
   }

   open(TCATOUTPUT,"<fort.12"); # obtain the current thickness of TPS file
   $_=<TCATOUTPUT>;
   /^\s*(\S*)/;
   $thickness=$1;
   close(TCATOUTPUT);

   $thkns[$loop_index]=$thickness;
```

```
   printf FINALOUTPUT "%8s - TPS thickness = %6.3f
in.\n",$name,$thickness*100/2.54;
}

#system("rm point*");

# ---------- end of heating analysis ----------------
$avg_thickness=0.0;
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   $avg_thickness=$avg_thickness+$thkns[$loop_index];
}
$avg_thickness=$avg_thickness/$end;


open(DEN,"<material_density"); # obtain the density of TPS material
$_=<DEN>;
/^\s*(\S*)/;
$tps_density=$1;
close(DEN);

if (($sharp eq "yes") && ($bodyside eq "windward")) {
   $sharp_weight=4.6e-3*$cowlarea;     #SHARP weight in lbm
   if ($material eq "RCC_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1577.8347)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "RCC_Al") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1577.8347)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "SiC_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*2400.00)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "SiC_Al") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*2400.00)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "TUFI_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1313.53)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   else {

$tile_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2808
;      #obtain density of tps
   }
   $tile_area_to_body_area=$cowltile;
   $cowl_length=$cowl_length*100./2.54;
```

```
   printf FINALOUTPUT "\n";
   printf FINALOUTPUT "Chosen TPS material was $material\n\n";
   printf FINALOUTPUT "Cowl SHARP TPS: %7.4f lbm/ft\n",
$sharp_weight/$cowl_length;
   printf FINALOUTPUT "$material unit weight: %6.2f lbm/ft2 \n
",$tile_unit_weight;
   printf FINALOUTPUT "$material TPS area to body area ratio: %6.2f \n
",$tile_area_to_body_area;

}
if (($sharp eq "yes") && ($bodyside eq "leeward")) {
   $sharp_weight=4.6e-3*$cowlarea;     #SHARP weight in lbm

$blanket_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2
808;  #obtain density of tps
   $blanket_area_to_body_area=$cowlblanket;
   $cowl_length=$cowl_length*100./2.54;
   printf FINALOUTPUT "\n";
   printf FINALOUTPUT "Chosen TPS material was $material\n\n";
   printf FINALOUTPUT "Cowl SHARP TPS: %6.2f lbm/ft\n",
$sharp_weight/$cowl_length;
   printf FINALOUTPUT "$material unit weight: %6.2f lbm/ft2 \n
",$blanket_unit_weight;
   printf FINALOUTPUT "$material TPS area to body area ratio: %6.2f \n
",$blanket_area_to_body_area;
}
if (($sharp eq "no") && ($bodyside eq "windward")) {
   if ($material eq "RCC_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1577.8347)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
      #obtain density of tps
   }
   elsif ($material eq "RCC_Al") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1577.8347)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
      #obtain density of tps
   }
   elsif ($material eq "SiC_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*2400.00)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
      #obtain density of tps
   }
   elsif ($material eq "SiC_Al") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*2400.00)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
      #obtain density of tps
   }
   elsif ($material eq "TUFI_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1313.53)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
      #obtain density of tps
   }
   else {
```

```
$tile_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2808
;      #obtain density of tps
   }
   $tile_area_to_body_area=$cowltile;
   printf FINALOUTPUT "Chosen TPS material was $material\n\n";
   printf FINALOUTPUT "$material unit weight: %6.2f lbm/ft2 \n
",$tile_unit_weight;
   printf FINALOUTPUT "$material TPS area to body area ratio: %6.2f \n
",$tile_area_to_body_area;
}
if (($sharp eq "no") && ($bodyside eq "leeward")) {

$blanket_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2
808;  #obtain density of tps
   $blanket_area_to_body_area=$cowlblanket;
   printf FINALOUTPUT "Chosen TPS material was $material\n\n";
   printf FINALOUTPUT "$material unit weight: %6.2f lbm/ft2 \n
",$blanket_unit_weight;
   printf FINALOUTPUT "$material TPS area to body area ratio: %6.2f \n
",$blanket_area_to_body_area;
}
close(FINALOUTPUT);

#----------------------- print needed information to website ------------------
------------
#
# print information to webpage "tcat_output.html"
#
open(OUT,"<thickness.txt");            #opens the input file for TCAT script
open(WEB,"<tcat_output.html");   #open tcat html file
print "Content-type:text/html\n\n";
while(<WEB>){
   if(/Insert stuff here/){
      print "<BR>\n";
      while(<OUT>){
         print "<BR>\n";
        print $_;
      }
   }
   else{
      print $_;
   }
}
close(WEB);
close(OUT);
```

```
#!/usr/sbin/perl

# Author: Kris Cowart
# Date Created: 8 Feb 00
#------------------------ Obtain Information From Web Browser-------------------
-----------
#
# This section of code allows a cgi script to run from a web browser.
# The information is passed using the POST and the input information is
# parceled out in a mesh type variable and stores in $FORM{$name}.
#

read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
       ($name, $value) = split(/=/, $pair);
       $value =~ tr/+/ /;
       $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
       $value =~ s/~!/ ~!/g;
       $FORM{$name} = $value;
}

#-------- assign information from input website to appropriate variable -------

$wingarea=$FORM{'wingarea'};
$wingtile=$FORM{'wingtile'};
$wingblanket=$FORM{'wingblanket'};
$sharp=$FORM{'sharp'};
$wing_length=$FORM{'leadingedge'};
$filename=$FORM{'filename'};
$bodyside=$FORM{'side'};
$material=$FORM{'tps'};

#------------------ MINIVER file is partitioned into body point files ---------
-----------
#
# partition the MINIVER file into separate body point files that includes time,
heat rate, temp
#
$i=0;                          #initialization of counter index "i"
$filecount=0;
open(FILE,"<$filename");       #open statement with filehandle FILE

until(eof FILE) {              #goes through the MINVER file until the end is
reached
   START: $_=<FILE>;           #skips the first line of the data file
   if (/^.{73}(\S*)/) {               #searches for the bodypoint label
   $title=$1;

   system("touch point$title");       #create body point file
   system("chmod 777 point$title");   #change permissions inorder to write to
body point file

   $filecount=$filecount+1;
   } # end of if statement

   while(<FILE>) {                    #while loop that reads thru each line of
the file
      $i=$i+1;
```

```
/^\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S
*)\s*(\S*)\s*(\S*)\s*(\S*)/;
        $time[$i]=$1;              #array declaration for time
        $tradeq[$i]=$9;                    #array declaration for rad eq temp
        $qconv[$i]=$10;                    #array declaration for conv heat rate
        $n=$i;
        if ($1==-100) {
           open(NEWFILE,">point$title"); #open statement for new file "newsample"
          $timesteps=$n-1;
           printf NEWFILE "$timesteps \n";#formatted print statement"
          for($i=1; $i<$n; $i++) {
               printf NEWFILE "%f \n",$time[$i];
               printf NEWFILE "%f \n",$qconv[$i];
               printf NEWFILE "%f \n",$tradeq[$i];
           } #end of for loop

           close(NEWFILE);
           $i=0;
           goto START
        } #end of if statement

    } #end of while loop
} #end of until loop
close(FILE);
#--------------------------
# incrementally move each body point file into "miniver.in" and execute tcat
# perl script that creates the filenames needed inorder to batch execute tcat
# filename are created via input from the user
# inputs (file_start_number, file_end_number, body_side)

$start=1;               #start of the file index
$end=($filecount-1)/2;     #end of the file index
#$end=1;

open(FILE,">inputs_for_outputs");
printf FILE "$bodyside\n";
printf FILE "$filename\n";
printf FILE "$end\n";
close(FILE);

system("touch thickness.txt");
system("chmod 777 thickness.txt");
open(FINALOUTPUT,">thickness.txt");
print FINALOUTPUT "TPS design results for $bodyside analysis for $filename
\n\n";
#-------------
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {

    if ($bodyside eq "leeward") {
       $name="pointl$loop_index";
    }
    if ($bodyside eq "windward") {
       $name="pointw$loop_index";
    }
    system("touch /home/asdl1/kcowart/public_html/miniver.in");
    system("chmod 777 /home/asdl1/kcowart/public_html/miniver.in");
    `cp /home/asdl1/kcowart/public_html/$name
/home/asdl1/kcowart/public_html/miniver.in`;
```

```
# ---------------- assign material properties ------------------------------
   if ($material eq "AETB8_Al") {
      $inputfile="AETB_8_AL_5inputs.in";
   }
   if ($material eq "AETB8_GrEx") {
      $inputfile="AETB_8_GrEx_5inputs.in";
   }
   if ($material eq "AETB12_Al") {
      $inputfile="AETB_12_AL_5inputs.in";
   }
   if ($material eq "AETB12_GrEx") {
      $inputfile="AETB_12_GrEx_5inputs.in";
   }
   if ($material eq "FRCI12_Al") {
      $inputfile="FRCI_12_AL_5inputs.i";
   }
   if ($material eq "FRCI12_GrEx") {
      $inputfile="FRCI_12_GrEx_5inputs.in";
   }
   if ($material eq "FRCI20_Al") {
      $inputfile="FRCI_20_AL_5inputs.in";
   }
   if ($material eq "FRCI20_GrEx") {
      $inputfile="FRCI_20_GrEx_5inputs.in";
   }
   if ($material eq "LI900_Al") {
      $inputfile="LI_900_AL_5inputs.in";
   }
   if ($material eq "LI900_GrEx") {
      $inputfile="LI_900_GrEx_5inputs.in";
   }
   if ($material eq "LI2200_Al") {
      $inputfile="LI_2200_AL_5inputs.in";
   }
   if ($material eq "LI2200_GrEx") {
      $inputfile="LI_2200_GrEx_5inputs.in";
   }
   if ($material eq "RCC_GrEx") {
      $inputfile="RCC_GrEx_5inputs.in";
   }
   if ($material eq "RCC_Al") {
      $inputfile="RCC_AL_5inputs.in";
   }
   if ($material eq "SiC_GrEx") {
      $inputfile="SIC_GrEx_5inputs.in";
   }
   if ($material eq "SiC_Al") {
      $inputfile="SIC_AL_5inputs.in";
   }
   if ($material eq "TUFI_GrEx") {
      $inputfile="TUFI_GrEx_5inputs.in";
   }
   if ($material eq "AFRSI_Al") {
      $inputfile="AFRSI_Al_3inputs.in";
   }
   if ($material eq "AFRSI_GrEx") {
      $inputfile="AFRSI_GrEx_3inputs.in";
   }
      if ($material eq "CFBI_Al") {
      $inputfile="CFBI_Al_3inputs.in";
```

```
   }
      if ($material eq "CFBI_GrEx") {
      $inputfile="CFBI_GrEx_3inputs.in";
   }
      if ($material eq "DURAFRSI_Al") {
      $inputfile="DURAFRSI_Al_3inputs.in";
   }
      if ($material eq "DURAFRSI_GrEx") {
      $inputfile="DURAFRSI_GrEx_3inputs.in";
   }
      if ($material eq "PBI_Al") {
      $inputfile="PBI_Al_3inputs.in";
   }
      if ($material eq "PBI_GrEx") {
      $inputfile="PBI_GrEx_3inputs.in";
   }
      if ($material eq "TABI_Al") {
      $inputfile="TABI_Al_3inputs.in";
   }
      if ($material eq "TABI_GrEx") {
      $inputfile="TABI_GrEx_3inputs.in";
   }
# -------------
   system("touch inputs.in");
   system("chmod 777 inputs.in");
   system("cp $inputfile inputs.in >> error 2>&1");

   if ($material eq "RCC_GrEx") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material eq "RCC_Al") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material eq "SiC_GrEx") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material eq "SiC_Al") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material eq "TUFI_GrEx") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   else {
      system("/home/asdl1/kcowart/public_html/go.exe>junkoutput");# run TCAT
   }

   open(TCATOUTPUT,"<fort.12"); # obtain the current thickness of TPS file
   $_=<TCATOUTPUT>;
   /^\s*(\S*)/;
   $thickness=$1;
   close(TCATOUTPUT);

   $thkns[$loop_index]=$thickness;
```

```
   printf FINALOUTPUT "%8s - TPS thickness = %6.3f
in.\n",$name,$thickness*100/2.54;
}

# --------- end of heating analysis ---------------
$avg_thickness=0.0;
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   $avg_thickness=$avg_thickness+$thkns[$loop_index];
}
$avg_thickness=$avg_thickness/$end;



open(DEN,"<material_density"); # obtain the density of TPS material
$_=<DEN>;
/^\s*(\S*)/;
$tps_density=$1;
close(DEN);

if (($sharp eq "yes") && ($bodyside eq "windward")) {
   $sharp_weight=4.6e-3*$wingarea;      #SHARP weight in lbm
   if ($material eq "RCC_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1577.8347)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "RCC_Al") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1577.8347)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "SiC_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*2400.00)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "SiC_Al") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*2400.00)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "TUFI_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1313.53)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   else {

$tile_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2808
;      #obtain density of tps
   }
   $tile_area_to_body_area=$wingtile;
   $wing_length=$wing_length*100./2.54;
   printf FINALOUTPUT "\n";
   printf FINALOUTPUT "Chosen TPS material was $material\n\n";
```

```
   printf FINALOUTPUT "Wing SHARP TPS: %7.4f lbm/ft\n",
$sharp_weight/$wing_length;
   printf FINALOUTPUT "$material unit weight: %6.2f lbm/ft2 \n
",$tile_unit_weight;
   printf FINALOUTPUT "$material TPS area to body area ratio: %6.2f \n
",$tile_area_to_body_area;
}
if (($sharp eq "yes") && ($bodyside eq "leeward")) {
   $sharp_weight=4.6e-3*$wingarea;     #SHARP weight in lbm

$blanket_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2
808;  #obtain density of tps
   $blanket_area_to_body_area=$wingblanket;
   $wing_length=$wing_length*100./2.54;
   printf FINALOUTPUT "\n";
   printf FINALOUTPUT "Chosen TPS material was $material\n\n";
   printf FINALOUTPUT "Wing SHARP TPS: %6.2f lbm/ft\n",
$sharp_weight/$wing_length;
   printf FINALOUTPUT "$material unit weight: %6.2f lbm/ft2 \n
",$blanket_unit_weight;
   printf FINALOUTPUT "$material TPS area to body area ratio: %6.2f \n
",$blanket_area_to_body_area;
}
if (($sharp eq "no") && ($bodyside eq "windward")) {
   if ($material eq "RCC_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1577.8347)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "RCC_Al") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1577.8347)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "SiC_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*2400.00)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "SiC_Al") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*2400.00)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   elsif ($material eq "TUFI_GrEx") {

$tile_unit_weight=($avg_thickness+0.00254)*3.2808*($avg_thickness*$tps_density+0
.00254*1313.53)/(0.00254+$avg_thickness)/0.4536/3.2808/3.2808/3.2808;
       #obtain density of tps
   }
   else {

$tile_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2808
;      #obtain density of tps
   }
   $tile_area_to_body_area=$wingtile;
```

```
   printf FINALOUTPUT "Chosen TPS material was $material\n\n";
   printf FINALOUTPUT "$material unit weight: %6.2f lbm/ft2 \n
",$tile_unit_weight;
   printf FINALOUTPUT "$material TPS area to body area ratio: %6.2f \n
",$tile_area_to_body_area;
}
if (($sharp eq "no") && ($bodyside eq "leeward")) {

$blanket_unit_weight=$avg_thickness*3.2808*$tps_density/0.4536/3.2808/3.2808/3.2
808;  #obtain density of tps
   $blanket_area_to_body_area=$wingblanket;
   printf FINALOUTPUT "Chosen TPS material was $material\n\n";
   printf FINALOUTPUT "$material unit weight: %6.2f lbm/ft2 \n
",$blanket_unit_weight;
   printf FINALOUTPUT "$material TPS area to body area ratio: %6.2f \n
",$blanket_area_to_body_area;
}
close(FINALOUTPUT);

#----------------------- print needed information to website ------------------
------------
#
# print information to webpage "tcat_output.html"
#
open(OUT,"<thickness.txt");           #opens the input file for TCAT script
open(WEB,"<tcat_output.html");  #open tcat html file
print "Content-type:text/html\n\n";
while(<WEB>){
   if(/Insert stuff here/){
      print "<BR>\n";
      while(<OUT>){
         print "<BR>\n";
        print $_;
      }
   }
   else{
      print $_;
   }
}
close(WEB);
close(OUT);
```

APPENDIX D

TCAT MULTIPLE TPS DESIGN CGI SCRIPTS

```
#!/usr/sbin/perl

# --------------------------------------------------------
# Obtain Information From Web Browser
# --------------------------------------------------------
#
# This section of code allows a cgi script to run from a web browser.
# The information is passed using the POST and the input information is
# parceled out in a mesh type variable and stores in $FORM{$name}.
#

read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
       ($name, $value) = split(/=/, $pair);
       $value =~ tr/+/ /;
       $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
       $value =~ s/~!/ ~!/g;
       $FORM{$name} = $value;
}

# ------------------------------------------------------------
# assign information from input website to appropriate variable
# ------------------------------------------------------------

$bodyarea=$FORM{'bodyarea'};
$tileratio=$FORM{'tileratio'};
$blanketratio=$FORM{'blanketratio'};
$bodyside=$FORM{'side'};
$backfacematerial=$FORM{'backface'};
$sharp=$FORM{'sharp'};
$tps_family=$FORM{'tpsfamily'};
$filename=$FORM{'filename'};

# -------------------------------------------------------------------------
# create the body point files
# -------------------------------------------------------------------------

$i=0;                            #initialization of counter index "i"
$filecount=0;
open(FILE,"<$filename");         #open statement with filehandle FILE

until(eof FILE) {                #goes through the MINVER file until the end is
reached
   START: $_=<FILE>;             #skips the first line of the data file

   if (/^.{73}(\S*)/) {                #searches for the bodypoint label
   $title=$1;
   $filecount=$filecount+1;
   } # end of if statement

   while(<FILE>) {                     #while loop that reads thru each line of
the file
       $i=$i+1;

/^\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S
*)\s*(\S*)\s*(\S*)\s*(\S*)/;
       $time[$i]=$1;             #array declaration for time
       $tradeq[$i]=$9;                 #array declaration for rad eq temp
```

```
      $qconv[$i]=$10;                    #array declaration for conv heat rate
      $n=$i;
      if ($1==-100) {
         open(NEWFILE,">point$title"); #open statement for new file "newsample"
        $timesteps=$n-1;
         printf NEWFILE "$timesteps \n";#formatted print statement"
        for($i=1; $i<$n; $i++) {
            printf NEWFILE "%f \n",$time[$i];
            printf NEWFILE "%f \n",$qconv[$i];
            printf NEWFILE "%f \n",$tradeq[$i];
         } #end of for loop

         close(NEWFILE);
         $i=0;
         goto START
      } #end of if statement

   } #end of while loop
} #end of until loop
close(FILE);

# ----------------------------------------------------------------------------
# loop trough the body point files and determine the materials for each
# body point
# ----------------------------------------------------------------------------

$start=1;             #start of the file index
$end=($filecount-1)/2;    #end of the file index
#$end=1;

open(FILE,">inputs_for_outputs");
printf FILE "$bodyside\n";
printf FILE "$filename\n";
printf FILE "$end\n";
close(FILE);

# -------------------------------------------
# determine the TPS based on the Rad Eq Temp
# -------------------------------------------

for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   if ($bodyside eq "leeward") {
      $name[$loop_index]="pointl$loop_index"; # stores the bodypoint filename
   }
   if ($bodyside eq "windward") {
      $name[$loop_index]="pointw$loop_index"; # stores the bodypoint filename
   }

# ------------------------------------------------
# read inforamtion from body point files
# ------------------------------------------------

   $maxradeqtemp = 0.0;
   open(FILE,"< $name[$loop_index]");
      $k = readline(*FILE);
      chop($k);
      for ($index = 1; $index <= $k; $index++) {
         $time[$index] = readline(*FILE);
         chop($time[$index]);
         $qconv[$index] = readline(*FILE);
```

```
        chop($qconv[$index]);
        $tradeq[$index] = readline(*FILE);
        chop($tradeq[$index]);
    }
  close(FILE);

# ---------------------------------------------------------------------------
-
# dynamically allowcate files for Radiation Equilibrium Temp for each body point
# ---------------------------------------------------------------------------
-

  open(FILE,"> $name[$loop_index]_radeqtemp");
  for ($index=1; $index <= $k; $index++) {
     printf FILE "%10.2f   %10.2f\n",$time[$index],$tradeq[$index];
  }
  close(FILE);

# -----------------------------------------------------
# find the maximum value of the rastemperature
# -----------------------------------------------------

  $i = 0;
  $maxradeqtemp=$tradeq[1];
  for($i = 2; $i <= $n; $i++) {
     if ($tradeq[$i] > $maxradeqtemp) {
        $maxradeqtemp=$tradeq[$i];
     }
  }

# ------------------------------------------------------------------------
# make material selection based on backface material,  side of vehicle, and
# rad eq temp, and conduct heating analysis at each body point
# ------------------------------------------------------------------------

if ($tps_family eq "all") {
  if ($backfacematerial eq "TiAl") {
     if ($bodyside eq "windward") {
        if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
           $material[$loop_index] = "RCC_tiles";
           $inputfile[$loop_index]="RCC_TiAl_5inputs.in";
          $score[$loop_index]=5;
        }
        elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1699.667)) {
           $material[$loop_index] = "SiC_tiles";
           $inputfile[$loop_index]="SIC_TiAl_5inputs.in";
          $score[$loop_index]=4;
        }
        elsif (($maxradeqtemp <= 1699.667) and ($maxradeqtemp > 1644.111)) {
           $material[$loop_index] = "AETB12_tiles";
           $inputfile[$loop_index]="AETB_12_TiAl_5inputs.in";
          $score[$loop_index]=3;
        }
        elsif (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 1588.556)) {
           $material[$loop_index] = "AETB8_tiles";
           $inputfile[$loop_index]="AETB_8_TiAl_5inputs.in";
          $score[$loop_index]=2;
        }
        elsif (($maxradeqtemp <= 1588.556) and ($maxradeqtemp > 0.0)) {
           $material[$loop_index] = "LI900_tiles";
```

```
        $inputfile[$loop_index]="LI_900_TiAl_5inputs.in";
       $score[$loop_index]=1;
      }
   }
   elsif ($bodyside eq "leeward") {
      if(($maxradeqtemp <= 1477.444) and ($maxradeqtemp > 1366.333)) {
         $material[$loop_index] = "CFBI_blankets";
         $inputfile[$loop_index]="CFBI_TiAl_3inputs.in";
       $score[$loop_index]=7;
      }
      elsif (($maxradeqtemp <= 1366.333) and ($maxradeqtemp > 1310.778)) {
         $material[$loop_index] = "AFRSI2500_blankets";
         $inputfile[$loop_index]="AFRSI2500_TiAl_3inputs.in";
       $score[$loop_index]=6;
      }
      elsif (($maxradeqtemp <= 1310.778) and ($maxradeqtemp > 1255.222)) {
         $material[$loop_index] = "AFRSI2200_blankets";
         $inputfile[$loop_index]="AFRSI2200_TiAl_3inputs.in";
       $score[$loop_index]=5;
      }
      elsif (($maxradeqtemp <= 1255.222) and ($maxradeqtemp > 921.889)) {
         $material[$loop_index] = "DURAFRSI_blankets";
         $inputfile[$loop_index]="DURAFRSI_TiAl_3inputs.in";
       $score[$loop_index]=4;
      }
      elsif (($maxradeqtemp <= 921.889) and ($maxradeqtemp > 699.667)) {
         $material[$loop_index] = "AFRSI_blankets";
         $inputfile[$loop_index]="AFRSI_TiAl_3inputs.in";
       $score[$loop_index]=3;
      }
      elsif (($maxradeqtemp <= 699.667) and ($maxradeqtemp > 505.222)) {
         $material[$loop_index] = "PBI_blankets";
         $inputfile[$loop_index]="PBI_TiAl_3inputs.in";
       $score[$loop_index]=2;
      }
      elsif (($maxradeqtemp <= 505.222) and ($maxradeqtemp > 0.0)) {
         $material[$loop_index] = "FRSI_blankets";
         $inputfile="FRSI_TiAl_3inputs.in";
       $score[$loop_index]=1;
      }
   }
}
elsif ($backfacematerial eq "GrEx") {
   if ($bodyside eq "windward") {
      if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
         $material[$loop_index] = "RCC_tiles";
         $inputfile[$loop_index]="RCC_GrEx_5inputs.in";
       $score[$loop_index]=5;
      }
      elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1699.667)) {
         $material[$loop_index] = "SiC_tiles";
         $inputfile[$loop_index]="SIC_GrEx_5inputs.in";
       $score[$loop_index]=4;
      }
      elsif (($maxradeqtemp <= 1699.667) and ($maxradeqtemp > 1644.111)) {
         $material[$loop_index] = "AETB12_tiles";
         $inputfile[$loop_index]="AETB_12_GrEx_5inputs.in";
       $score[$loop_index]=3;
      }
      elsif (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 1588.556)) {
```

```
            $material[$loop_index] = "AETB8_tiles";
            $inputfile[$loop_index]="AETB_8_GrEx_5inputs.in";
           $score[$loop_index]=2;
          }
          elsif (($maxradeqtemp <= 1588.556) and ($maxradeqtemp > 0.0)) {
            $material[$loop_index] = "LI900_tiles";
            $inputfile[$loop_index]="LI_900_GrEx_5inputs.in";
           $score[$loop_index]=1;
          }
        }
        elsif ($bodyside eq "leeward") {
          if(($maxradeqtemp <= 1477.444) and ($maxradeqtemp > 1366.333)) {
            $material[$loop_index] = "CFBI_blankets";
            $inputfile[$loop_index]="CFBI_GrEx_3inputs.in";
           $score[$loop_index]=7;
          }
          elsif (($maxradeqtemp <= 1366.333) and ($maxradeqtemp > 1310.778)) {
            $material[$loop_index] = "AFRSI2500_blankets";
            $inputfile[$loop_index]="AFRSI2500_GrEx_3inputs.in";
           $score[$loop_index]=6;
          }
          elsif (($maxradeqtemp <= 1310.778) and ($maxradeqtemp > 1255.222)) {
            $material[$loop_index] = "AFRSI2200_blankets";
            $inputfile[$loop_index]="AFRSI2200_GrEx_3inputs.in";
           $score[$loop_index]=5;
          }
          elsif (($maxradeqtemp <= 1255.222) and ($maxradeqtemp > 921.889)) {
            $material[$loop_index] = "DURAFRSI_blankets";
            $inputfile[$loop_index]="DURAFRSI_GrEx_3inputs.in";
           $score[$loop_index]=4;
          }
          elsif (($maxradeqtemp <= 921.889) and ($maxradeqtemp > 699.667)) {
            $material[$loop_index] = "AFRSI_blankets";
            $inputfile[$loop_index]="AFRSI_GrEx_3inputs.in";
           $score[$loop_index]=3;
          }
          elsif (($maxradeqtemp <= 699.667) and ($maxradeqtemp > 505.222)) {
            $material[$loop_index] = "PBI_blankets";
            $inputfile[$loop_index]="PBI_GrEx_3inputs.in";
           $score[$loop_index]=2;
          }
          elsif (($maxradeqtemp <= 505.222) and ($maxradeqtemp > 0.0)) {
            $material[$loop_index] = "FRSI_blankets";
            $inputfile[$loop_index]="FRSI_GrEx_3inputs.in";
           $score[$loop_index]=1;
          }
        }
      }
    }
    elsif ($tps_family eq "nextgen") {
      if ($backfacematerial eq "TiAl") {
        if ($bodyside eq "windward") {
          if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
            $material[$loop_index] = "RCC_tiles";
            $inputfile[$loop_index]="RCC_TiAl_5inputs.in";
           $score[$loop_index]=3;
          }
          elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1588.556)) {
            $material[$loop_index] = "SiC_tiles";
            $inputfile[$loop_index]="SIC_TiAl_5inputs.in";
```

```
          $score[$loop_index]=2;
        }
        elsif (($maxradeqtemp <= 1588.556) and ($maxradeqtemp > 200.00)) {
           $material[$loop_index] = "LI900_tiles";
           $inputfile[$loop_index]="LI_900_TiAl_5inputs.in";
          $score[$loop_index]=1;
        }
      }
      elsif ($bodyside eq "leeward") {
        if(($maxradeqtemp <= 1477.444) and ($maxradeqtemp > 1366.333)) {
           $material[$loop_index] = "CFBI_blankets";
           $inputfile[$loop_index]="CFBI_TiAl_3inputs.in";
          $score[$loop_index]=4;
        }
        elsif (($maxradeqtemp <= 1366.333) and ($maxradeqtemp > 1310.778)) {
           $material[$loop_index] = "AFRSI2500_blankets";
           $inputfile[$loop_index]="AFRSI2500_TiAl_3inputs.in";
          $score[$loop_index]=3;
        }
        elsif (($maxradeqtemp <= 1310.778) and ($maxradeqtemp > 1255.222)) {
           $material[$loop_index] = "AFRSI2200_blankets";
           $inputfile[$loop_index]="AFRSI2200_TiAl_3inputs.in";
          $score[$loop_index]=2;
        }
        elsif (($maxradeqtemp <= 1255.222) and ($maxradeqtemp > 200.00)) {
           $material[$loop_index] = "DURAFRSI_blankets";
           $inputfile[$loop_index]="DURAFRSI_TiAl_3inputs.in";
          $score[$loop_index]=1;
        }
      }
    }
    elsif ($backfacematerial eq "GrEx") {
      if ($bodyside eq "windward") {
        if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
           $material[$loop_index] = "RCC_tiles";
           $inputfile[$loop_index]="RCC_GrEx_5inputs.in";
          $score[$loop_index]=3;
        }
        elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1588.556)) {
           $material[$loop_index] = "SiC_tiles";
           $inputfile[$loop_index]="SIC_GrEx_5inputs.in";
          $score[$loop_index]=2;
        }
        elsif (($maxradeqtemp <= 1588.556) and ($maxradeqtemp > 200.00)) {
           $material[$loop_index] = "LI900_tiles";
           $inputfile[$loop_index]="LI_900_GrEx_5inputs.in";
          $score[$loop_index]=1;
        }
      }
      elsif ($bodyside eq "leeward") {
        if(($maxradeqtemp <= 1477.444) and ($maxradeqtemp > 1366.333)) {
           $material[$loop_index] = "CFBI_blankets";
           $inputfile[$loop_index]="CFBI_GrEx_3inputs.in";
          $score[$loop_index]=4;
        }
        elsif (($maxradeqtemp <= 1366.333) and ($maxradeqtemp > 1310.778)) {
           $material[$loop_index] = "AFRSI2500_blankets";
           $inputfile[$loop_index]="AFRSI2500_GrEx_3inputs.in";
          $score[$loop_index]=3;
        }
```

```
         elsif (($maxradeqtemp <= 1310.778) and ($maxradeqtemp > 1255.222)) {
            $material[$loop_index] = "AFRSI2200_blankets";
            $inputfile[$loop_index]="AFRSI2200_GrEx_3inputs.in";
           $score[$loop_index]=2;
         }
         elsif (($maxradeqtemp <= 1255.222) and ($maxradeqtemp > 200.00)) {
            $material[$loop_index] = "DURAFRSI_blankets";
            $inputfile[$loop_index]="DURAFRSI_GrEx_3inputs.in";
           $score[$loop_index]=1;
         }
      }
   }
}
elsif ($tps_family eq "shuttle") {
   if ($backfacematerial eq "TiAl") {
      if ($bodyside eq "windward") {
         if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
            $material[$loop_index] = "RCC_tiles";
            $inputfile[$loop_index]="RCC_TiAl_5inputs.in";
           $score[$loop_index]=3;
         }
         elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1644.111)) {
            $material[$loop_index] = "LI2200_tiles";
            $inputfile[$loop_index]="LI_2200_TiAl_5inputs.in";
           $score[$loop_index]=2;
         }
         elsif (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 200.0)) {
            $material[$loop_index] = "FRCI_tiles";
            $inputfile[$loop_index]="FRCI_12_TiAl_5inputs.in";
           $score[$loop_index]=1;
         }
      }
      elsif ($bodyside eq "leeward") {
         if (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 921.889)) {
            $material[$loop_index] = "FRCI_tiles";
            $inputfile[$loop_index]="FRCI_12_TiAl_5inputs.in";
           $score[$loop_index]=3;
         }
         if (($maxradeqtemp <= 921.889) and ($maxradeqtemp > 505.222)) {
            $material[$loop_index] = "AFRSI_blankets";
            $inputfile[$loop_index]="AFRSI_TiAl_3inputs.in";
           $score[$loop_index]=2;
         }
         elsif (($maxradeqtemp <= 505.222) and ($maxradeqtemp > 200.0)) {
            $material[$loop_index] = "FRSI_blankets";
            $inputfile="FRSI_TiAl_3inputs.in";
           $score[$loop_index]=1;
         }
      }
   }
   elsif ($backfacematerial eq "GrEx") {
      if ($bodyside eq "windward") {
         if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
            $material[$loop_index] = "RCC_tiles";
            $inputfile[$loop_index]="RCC_GrEx_5inputs.in";
           $score[$loop_index]=3;
         }
         elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1644.111)) {
            $material[$loop_index] = "LI2200_tiles";
            $inputfile[$loop_index]="LI_2200_GrEx_5inputs.in";
```

```
          $score[$loop_index]=2;
         }
         elsif (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 200.0)) {
            $material[$loop_index] = "FRCI_tiles";
            $inputfile[$loop_index]="FRCI_12_GrEx_5inputs.in";
          $score[$loop_index]=1;
         }
      }
      elsif ($bodyside eq "leeward") {
         if (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 921.889)) {
            $material[$loop_index] = "FRCI_tiles";
            $inputfile[$loop_index]="FRCI_12_GrEx_5inputs.in";
          $score[$loop_index]=3;
         }
         if (($maxradeqtemp <= 921.889) and ($maxradeqtemp > 505.222)) {
            $material[$loop_index] = "AFRSI_blankets";
            $inputfile[$loop_index]="AFRSI_GrEx_3inputs.in";
          $score[$loop_index]=2;
         }
         elsif (($maxradeqtemp <= 505.222) and ($maxradeqtemp > 200.0)) {
            $material[$loop_index] = "FRSI_blankets";
            $inputfile="FRSI_GrEx_3inputs.in";
          $score[$loop_index]=1;
         }
      }
   }
}
}

# ------------------------------------------------------------
# logic check to prevent patchwork of materials on RLV
# ------------------------------------------------------------

$RCC_count = 0;
$SIC_count = 0;
$AETB12_count = 0;
$AETB8_count = 0;
$LI900_count = 0;
$LI2200_count=0;
$FRCI_count=0;
$CFBI_count = 0;
$AFRSI2500_count = 0;
$AFRSI2200_count = 0;
$AFRSI_count = 0;
$PBI_count = 0;
$FRSI_count = 0;
$DURAFRSI_count=0;

$k=0;
for ($k = 3; $k <= $end; $k++) {
   if (($material[$k] eq $material[$k-2]) && ($material[$k] ne $material[$k-1])
&& ($score[$k] > $score[$k-1])) {
      $material[$k-1] = $material[$k];
      $inputfile[$k-1]=$inputfile[$k];
   }
   elsif (($material[$k] eq $material[$k-2]) && ($material[$k] ne $material[$k-
1]) && ($score[$k] < $score[$k-1])) {
      $material[$k-2] = $material[$k-1];
      $inputfile[$k-2]=$inputfile[$k-1];
      $material[$k] = $material[$k-1];
```

```
      $inputfile[$k]=$inputfile[$k-1];
   }
# ---------------------------------------------------------------------------
# A counter is placed on each one of the materials chosen.  This allows for the
# tps area to body area ratio to be calculated.
# ---------------------------------------------------------------------------
}
$k=0;
for ($k = 1; $k <= $end; $k++) {
   if ($material[$k] eq "RCC_tiles") {
      $RCC_count=$RCC_count+1;
   }
   elsif ($material[$k] eq "SiC_tiles") {
      $SIC_count=$SIC_count+1;
   }
   elsif ($material[$k] eq "AETB12_tiles") {
      $AETB12_count=$AETB12_count+1;
   }
   elsif ($material[$k] eq "AETB8_tiles") {
      $AETB8_count=$AETB8_count+1;
   }
   elsif ($material[$k] eq "LI900_tiles") {
      $LI900_count=$LI900_count+1;
   }
   elsif ($material[$k] eq "LI2200_tiles") {
      $LI2200_count=$LI2200_count+1;
   }
   elsif ($material[$k] eq "FRCI_tiles") {
      $FRCI_count=$FRCI_count+1;
   }
   elsif ($material[$k] eq "CFBI_blankets") {
      $CFBI_count=$CFBI_count+1;
   }
   elsif ($material[$k] eq "AFRSI2500_blankets") {
      $AFRSI2500_count=$AFRSI2500_count+1;
   }
   elsif ($material[$k] eq "AFRSI2200_blankets") {
      $AFRSI2200_count=$AFRSI2200_count+1;
   }
   elsif ($material[$k] eq "AFRSI_blankets") {
      $AFRSI_count=$AFRSI_count+1;
   }
   elsif ($material[$k] eq "DURAFRSI_blankets") {
      $DURAFRSI_count=$DURAFRSI_count+1;
   }
   elsif ($material[$k] eq "PBI_blankets") {
      $PBI_count=$PBI_count+1;
   }
   elsif ($material[$k] eq "FRSI_blankets") {
      $FRSI_count=$FRSI_count+1;
   }
}

# ------------------------------------------------
# conduct heating analysis
# ------------------------------------------------

open(FINALOUTPUT,">thickness.txt");
print FINALOUTPUT "TPS design results for $bodyside analysis for $filename \n";
```

```
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   system("touch /home/asdl1/kcowart/public_html/miniver.in");
   system("cp /home/asdl1/kcowart/public_html/$name[$loop_index]
/home/asdl1/kcowart/public_html/miniver.in>> error1 2>&1");
   system("touch inputs.in");
   system("cp $inputfile[$loop_index] inputs.in >> error2 2>&1");
   system("touch material_density");
   system("chmod 777 material_density");

   if ($material[$loop_index] eq "RCC_tiles") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material[$loop_index] eq "SiC_tiles") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   else {
      system("/home/asdl1/kcowart/public_html/go.exe>junkoutput");# run TCAT
   }

#   system("touch $name[$loop_index]_temphist");
#   system("chmod 777 $name[$loop_index]_temphist");
   system("mv fort.90 $name[$loop_index]_temphist");

#   system("touch $name[$loop_index]_heatratehist");
#   system("chmod 777 $name[$loop_index]_heatratehist");
   system("mv fort.91 $name[$loop_index]_heatratehist");

   open(TCATOUTPUT,"<fort.12"); # obtain thickness of current TPS material
   $_=<TCATOUTPUT>;
   /^\s*(\S*)/;
   $thickness[$loop_index]=$1;
   close(TCATOUTPUT);

   open(DEN,"<material_density"); # obtain the density of current TPS material
   $_=<DEN>;
   /^\s*(\S*)/;
   $tps_density[$loop_index]=$1;
   close(DEN);

# ------------------------------------------------------
# print the output to file thickness.txt
# ------------------------------------------------------

   printf FINALOUTPUT "%8s - %15s - TPS thickness = %8.2f
inches\n",$name[$loop_index],$material[$loop_index],$thickness[$loop_index]*(100
./2.54);
}
close(TEMP);

# ----------------------------------------------------------------------
# calculate the unit weight and average thickness
# ----------------------------------------------------------------------

$avg_thickness=0.0;
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   $avg_thickness=$avg_thickness+$thickness[$loop_index];
}
$avg_thickness=$avg_thickness/$end;
```

```
$avg_thickness=$avg_thickness*(100./2.54);

$unit_weights=0.;
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   $unit_weight=$unit_weight+$tps_density[$loop_index]*$thickness[$loop_index];
}
$unit_weight=($unit_weight/0.4536)/(3.2808**2);
$unit_weight=$unit_weight/$end;

$avg_unit_weight=$unit_weight/$end;

printf FINALOUTPUT "\n";
printf FINALOUTPUT "Average TPS thickness = %8.2f inches\n",$avg_thickness;
printf FINALOUTPUT "Smeared TPS unit_weight = %8.2f lbm/ft^2\n",$unit_weight;

# -------------------------------------------------------------------
# calculate the area percentages for each of the TPS materials used
# -------------------------------------------------------------------

if ($tps_family eq "all") {
   if ($bodyside eq "windward") {
      $RCC_ratio=$RCC_count/$end*$tileratio;
      $SIC_ratio=$SIC_count/$end*$tileratio;
      $AETB12_ratio=$AETB12_count/$end*$tileratio;
      $AETB8_ratio=$AETB8_count/$end*$tileratio;
      $LI900_ratio=$LI900_count/$end*$tileratio;
   }
   elsif ($bodyside eq "leeward") {
      $CFBI_ratio=$CFBI_count/$end*$blanketratio;
      $AFRSI2500_ratio=$AFRSI2500_count/$end*$blanketratio;
      $AFRSI2200_ratio=$AFRSI2200_count/$end*$blanketratio;
      $DURAFRSI_ratio=$DURAFRSI_count/$end*$blanketratio;
      $AFRSI_ratio=$AFRSI_count/$end*$blanketratio;
      $PBI_ratio=$PBI_count/$end*$blanketratio;
      $FRSI_ratio=$FRSI_count/$end*$blanketratio;
   }
}
elsif ($tps_family eq "nextgen") {
   if ($bodyside eq "windward") {
      $RCC_ratio=$RCC_count/$end*$tileratio;
      $SIC_ratio=$SIC_count/$end*$tileratio;
      $LI900_ratio=$LI900_count/$end*$tileratio;
   }
   elsif ($bodyside eq "leeward") {
      $CFBI_ratio=$CFBI_count/$end*$blanketratio;
      $AFRSI2500_ratio=$AFRSI2500_count/$end*$blanketratio;
      $AFRSI2200_ratio=$AFRSI2200_count/$end*$blanketratio;
      $DURAFRSI_ratio=$DURAFRSI_count/$end*$blanketratio;
   }
}
elsif ($tps_family eq "shuttle") {
   if ($bodyside eq "windward") {
      $RCC_ratio=$RCC_count/$end*$tileratio;
      $LI2200_ratio=$LI2200_count/$end*$tileratio;
      $FRCI_ratio=$FRCI_count/$end*$tileratio;
   }
   elsif ($bodyside eq "leeward") {
      $FRCI_ratio=$FRCI_count/$end*$blanketratio;
      $AFRSI_ratio=$AFRSI_count/$end*$blanketratio;
      $FRSI_ratio=$FRSI_count/$end*$blanketratio;
```

```
    }
}
$RCC_unitweight=0;
$SIC_unitweight=0;
$AETB12_unitweight=0;
$AETB8_unitweight=0;
$LI900_unitweight=0;
$LI2200_unitweight=0;
$FRCI_unitweight=0;
$CFBI_unitweight=0;
$AFRSI2500_unitweight=0;
$AFRSI2200_unitweight=0;
$AFRSI_unitweight=0;
$DURAFRSI_unitweight=0;
$PBI_unitweight=0;
$FRSI_unitweight=0;
$k=0;
for ($k = 1; $k <= $end; $k++) {
   if ($material[$k] eq "RCC_tiles") {

$RCC_unitweight=$RCC_unitweight+($tps_density[$k]*($thickness[$k]/(0.00254+$thic
kness[$k]))+(0.00254/(0.00254+$thickness[$k]))*1577.8347)*(0.00254+$thickness[$k
])/$RCC_count;
   }
   elsif ($material[$k] eq "SiC_tiles") {

$SIC_unitweight=$SIC_unitweight+($tps_density[$k]*($thickness[$k]/(0.00254+$thic
kness[$k]))+(0.00254/(0.00254+$thickness[$k]))*2400.00)*(0.00254+$thickness[$k])
/$SIC_count;
   }
   elsif ($material[$k] eq "AETB12_tiles") {

$AETB12_unitweight=$AETB12_unitweight+$tps_density[$k]*$thickness[$k]/$AETB12_co
unt;
   }
   elsif ($material[$k] eq "AETB8_tiles") {

$AETB8_unitweight=$AETB8_unitweight+$tps_density[$k]*$thickness[$k]/$AETB8_count
;
   }
   elsif ($material[$k] eq "LI900_tiles") {

$LI900_unitweight=$LI900_unitweight+$tps_density[$k]*$thickness[$k]/$LI900_count
;
   }
   elsif ($material[$k] eq "LI2200_tiles") {

$LI2200_unitweight=$LI2200_unitweight+$tps_density[$k]*$thickness[$k]/$LI2200_co
unt;
   }
   elsif ($material[$k] eq "FRCI_tiles") {

$FRCI_unitweight=$FRCI_unitweight+$tps_density[$k]*$thickness[$k]/$FRCI_count;
   }
   elsif ($material[$k] eq "CFBI_blankets") {

$CFBI_unitweight=$CFBI_unitweight+$tps_density[$k]*$thickness[$k]/$CFBI_count;
   }
   elsif ($material[$k] eq "AFRSI2500_blankets") {
```

```
$AFRSI2500_unitweight=$AFRSI2500_unitweight+$tps_density[$k]*$thickness[$k]/AFRS
I2500_count;
   }
   elsif ($material[$k] eq "AFRSI2200_blankets") {

$AFRSI2200_unitweight=$AFRSI2200_unitweight+$tps_density[$k]*$thickness[$k]/$AFR
SI2200_count;
   }
   elsif ($material[$k] eq "AFRSI_blankets") {

$AFRSI_unitweight=$AFRSI_unitweight+$tps_density[$k]*$thickness[$k]/$AFRSI_count
;
   }
   elsif ($material[$k] eq "DURAFRSI_blankets") {

$DURAFRSI_unitweight=$DURAFRSI_unitweight+$tps_density[$k]*$thickness[$k]/$DURAF
RSI_count;
   }
   elsif ($material[$k] eq "PBI_blankets") {

$PBI_unitweight=$PBI_unitweight+$tps_density[$k]*$thickness[$k]/$PBI_count;
   }
   elsif ($material[$k] eq "FRSI_blankets") {

$FRSI_unitweight=$FRSI_unitweight+$tps_density[$k]*$thickness[$k]/$FRSI_count;
   }
}

if (($sharp eq "yes") && ($bodyside eq "windward")) {
   $sharp_weight=4.6e-3*$bodyarea;      #SHARP weight in lbm
   $rcc_unit_weight=32.83;      #RCC unit weight
   $rcc_area_to_body_area=0.0075;

$tile_unit_weight=$avg_thickness*3.2808*$tps_density*(1/0.4536)*(1/3.2808)**3;
       #obtain density of tps
   $tile_area_to_body_area=$bodytile;
   printf FINALOUTPUT "\n";
   printf FINALOUTPUT "Nose SHARP TPS: %6.2f lbm\n",$sharp_weight;
}
if (($sharp eq "yes") && ($bodyside eq "leeward")) {
   $sharp_weight=4.6e-3*$bodyarea;      #SHARP weight in lbm
   printf FINALOUTPUT "\n";
   printf FINALOUTPUT "Nose SHARP TPS: %6.2f lbm\n",$sharp_weight;
}

if ($RCC_count > 0) {
   printf FINALOUTPUT "RCC unit weight: %6.2f lbm/ft2\n",
$RCC_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "RCC TPS Area to Body Area Ratio: %6.4f\n", $RCC_ratio;

}
if ($SIC_count > 0) {
   printf FINALOUTPUT "SiC unit weight: %6.2f lbm/ft2\n",
$SIC_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "SiC TPS Area to Body Area Ratio: %6.4f\n", $SIC_ratio;
}
if ($AETB12_count > 0) {
   printf FINALOUTPUT "AETB12 unit weight: %6.2f lbm/ft2\n",
$AETB12_unitweight/(0.4536*3.2808**2);
```

```
   printf FINALOUTPUT "AETB12 TPS Area to Body Area Ratio: %6.4f\n",
$AETB12_ratio;
}
if ($AETB8_count > 0) {
   printf FINALOUTPUT "AETB8 unit weight: %6.2f lbm/ft2\n",
$AETB8_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AETB8 TPS Area to Body Area Ratio: %6.4f\n",
$AETB8_ratio;
}
if ($LI900_count > 0) {
   printf FINALOUTPUT "LI900 unit weight: %6.2f lbm/ft2\n",
$LI900_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "LI900 TPS Area to Body Area Ratio: %6.4f\n",
$LI900_ratio;
}
if ($LI2200_count > 0) {
   printf FINALOUTPUT "LI2200 unit weight: %6.2f lbm/ft2\n",
$LI2200_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "LI2200 TPS Area to Body Area Ratio: %6.4f\n",
$LI2200_ratio;
}
if ($FRCI_count > 0) {
   printf FINALOUTPUT "FRCI unit weight: %6.2f lbm/ft2\n",
$FRCI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "FRCI TPS Area to Body Area Ratio: %6.4f\n", $FRCI_ratio;
}
if ($CFBI_count > 0) {
   printf FINALOUTPUT "CFBI unit weight: %6.2f lbm/ft2\n",
$CFBI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "CFBI TPS Area to Body Area Ratio: %6.4f\n", $CFBI_ratio;
}
if ($AFRSI2500_count > 0) {
   printf FINALOUTPUT "AFRSI2500 unit weight: %6.2f lbm/ft2\n",
$AFRSI2500_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AFRSI2500 TPS Area to Body Area Ratio: %6.4f\n",
$AFRSI2500_ratio;
}
if ($AFRSI2200_count > 0) {
   printf FINALOUTPUT "AFRSI2200 unit weight: %6.2f lbm/ft2\n",
$AFRSI2200_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AFRSI2200 TPS Area to Body Area Ratio: %6.4f\n",
$AFRSI2200_ratio;
}
if ($AFRSI_count > 0) {
   printf FINALOUTPUT "AFRSI unit weight: %6.2f lbm/ft2\n",
$AFRSI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AFRSI TPS Area to Body Area Ratio: %6.4f\n",
$AFRSI_ratio;
}
if ($DURAFRSI_count > 0) {
   printf FINALOUTPUT "DURAFRSI unit weight: %6.2f lbm/ft2\n",
$DURAFRSI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "DURAFRSI TPS Area to Body Area Ratio: %6.4f\n",
$DURAFRSI_ratio;
}
if ($PBI_count > 0) {
   printf FINALOUTPUT "PBI unit weight: %6.2f lbm/ft2\n",
$PBI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "PBI TPS Area to Body Area Ratio: %6.4f\n", $PBI_ratio;
}
```

```
if ($FRSI_count > 0) {
   printf FINALOUTPUT "FRSI unit weight: %6.2f lbm/ft2\n",
$FRSI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "FRSI TPS Area to Body Area Ratio: %6.4f\n", $FRSI_ratio;
}

close(FINALOUTPUT);


# ---------------------------------------------------------------------
# print desired output to webpage
# ---------------------------------------------------------------------

open(OUT,"<thickness.txt");            #opens the input file for TCAT script
open(WEB,"<tcat_output.html");   #open tcat html file
print "Content-type:text/html\n\n";
while(<WEB>){
   if(/Insert stuff here/){
      print "<BR>\n";
      while(<OUT>){
         print "<BR>\n";
        print $_;
      }
   }
   else{
     print $_;
   }
}
close(WEB);
close(OUT);
```

```
#!/usr/sbin/perl

# ----------------------------------------------------
# Obtain Information From Web Browser
# ----------------------------------------------------
#
# This section of code allows a cgi script to run from a web browser.
# The information is passed using the POST and the input information is
# parceled out in a mesh type variable and stores in $FORM{$name}.
#

read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
       ($name, $value) = split(/=/, $pair);
       $value =~ tr/+/ /;
       $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
       $value =~ s/~!/ ~!/g;
       $FORM{$name} = $value;
}

# ------------------------------------------------------------
# assign information from input website to appropriate variable
# ------------------------------------------------------------

$cowlarea=$FORM{'cowlarea'};
$tileratio=$FORM{'tileratio'};
$blanketratio=$FORM{'blanketratio'};
$cowllength=$FORM{'cowllength'};
$bodyside=$FORM{'side'};
$backfacematerial=$FORM{'backface'};
$sharp=$FORM{'sharp'};
$tps_family=$FORM{'tpsfamily'};
$filename=$FORM{'filename'};

# ---------------------------------------------------------------------------
# create the body point files
# ---------------------------------------------------------------------------

$i=0;                            #initialization of counter index "i"
$filecount=0;
open(FILE,"<$filename");         #open statement with filehandle FILE

until(eof FILE) {                #goes through the MINVER file until the end is
reached
   START: $_=<FILE>;             #skips the first line of the data file

   if (/^.{73}(\S*)/) {                #searches for the bodypoint label
   $title=$1;
   $filecount=$filecount+1;
   } # end of if statement

   while(<FILE>) {                     #while loop that reads thru each line of
the file
      $i=$i+1;

/^\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S
*)\s*(\S*)\s*(\S*)\s*(\S*)/;
      $time[$i]=$1;              #array declaration for time
```

```
    $tradeq[$i]=$9;                    #array declaration for rad eq temp
    $qconv[$i]=$10;                    #array declaration for conv heat rate
    $n=$i;
    if ($1==-100) {
       open(NEWFILE,">point$title"); #open statement for new file "newsample"
      $timesteps=$n-1;
       printf NEWFILE "$timesteps \n";#formatted print statement"
      for($i=1; $i<$n; $i++) {
          printf NEWFILE "%f \n",$time[$i];
          printf NEWFILE "%f \n",$qconv[$i];
          printf NEWFILE "%f \n",$tradeq[$i];
       } #end of for loop

       close(NEWFILE);
       $i=0;
       goto START
    } #end of if statement

  } #end of while loop
} #end of until loop
close(FILE);

# ---------------------------------------------------------------------------
# loop trough the body point files and determine the materials for each
# body point
# ---------------------------------------------------------------------------

$start=1;             #start of the file index
$end=($filecount-1)/2;    #end of the file index
#$end=1;

open(FILE,">inputs_for_outputs");
printf FILE "$bodyside\n";
printf FILE "$filename\n";
printf FILE "$end\n";
close(FILE);

# ------------------------------------------
# determine the TPS based on the Rad Eq Temp
# ------------------------------------------

for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   if ($bodyside eq "leeward") {
      $name[$loop_index]="pointl$loop_index"; # stores the bodypoint filename
   }
   if ($bodyside eq "windward") {
      $name[$loop_index]="pointw$loop_index"; # stores the bodypoint filename
   }

# ------------------------------------------------------
# read inforamtion from body point files
# ------------------------------------------------------

   $maxradeqtemp = 0.0;
   open(FILE,"< $name[$loop_index]");
      $k = readline(*FILE);
      chop($k);
      for ($index = 1; $index <= $k; $index++) {
         $time[$index] = readline(*FILE);
         chop($time[$index]);
```

```
        $qconv[$index] = readline(*FILE);
        chop($qconv[$index]);
        $tradeq[$index] = readline(*FILE);
        chop($tradeq[$index]);
      }
   close(FILE);

# ----------------------------------------------------------------------------
-
# dynamically allowcate files for Radiation Equilibrium Temp for each body point
# ----------------------------------------------------------------------------
-

   open(FILE,"> $name[$loop_index]_radeqtemp");
   for ($index=1; $index <= $k; $index++) {
      printf FILE "%10.2f   %10.2f\n",$time[$index],$tradeq[$index];
   }
   close(FILE);

# ---------------------------------------------------
# find the maximum value of the rastemperature
# ---------------------------------------------------

   $i = 0;
   $maxradeqtemp=$tradeq[1];
   for($i = 2; $i <= $n; $i++) {
      if ($tradeq[$i] > $maxradeqtemp) {
         $maxradeqtemp=$tradeq[$i];
      }
   }

# --------------------------------------------------------------------------
# make material selection based on backface material,  side of vehicle, and
# rad eq temp, and conduct heating analysis at each body point
# --------------------------------------------------------------------------

if ($tps_family eq "all") {
   if ($backfacematerial eq "TiAl") {
      if ($bodyside eq "windward") {
         if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
            $material[$loop_index] = "RCC_tiles";
            $inputfile[$loop_index]="RCC_TiAl_5inputs.in";
          $score[$loop_index]=5;
         }
         elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1699.667)) {
            $material[$loop_index] = "SiC_tiles";
            $inputfile[$loop_index]="SIC_TiAl_5inputs.in";
          $score[$loop_index]=4;
         }
         elsif (($maxradeqtemp <= 1699.667) and ($maxradeqtemp > 1644.111)) {
            $material[$loop_index] = "AETB12_tiles";
            $inputfile[$loop_index]="AETB_12_TiAl_5inputs.in";
          $score[$loop_index]=3;
         }
         elsif (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 1588.556)) {
            $material[$loop_index] = "AETB8_tiles";
            $inputfile[$loop_index]="AETB_8_TiAl_5inputs.in";
          $score[$loop_index]=2;
         }
         elsif (($maxradeqtemp <= 1588.556) and ($maxradeqtemp > 0.0)) {
```

```
            $material[$loop_index] = "LI900_tiles";
            $inputfile[$loop_index]="LI_900_TiAl_5inputs.in";
          $score[$loop_index]=1;
         }
      }
      elsif ($bodyside eq "leeward") {
         if(($maxradeqtemp <= 1477.444) and ($maxradeqtemp > 1366.333)) {
            $material[$loop_index] = "CFBI_blankets";
            $inputfile[$loop_index]="CFBI_TiAl_3inputs.in";
          $score[$loop_index]=7;
         }
         elsif (($maxradeqtemp <= 1366.333) and ($maxradeqtemp > 1310.778)) {
            $material[$loop_index] = "AFRSI2500_blankets";
            $inputfile[$loop_index]="AFRSI2500_TiAl_3inputs.in";
          $score[$loop_index]=6;
         }
         elsif (($maxradeqtemp <= 1310.778) and ($maxradeqtemp > 1255.222)) {
            $material[$loop_index] = "AFRSI2200_blankets";
            $inputfile[$loop_index]="AFRSI2200_TiAl_3inputs.in";
          $score[$loop_index]=5;
         }
         elsif (($maxradeqtemp <= 1255.222) and ($maxradeqtemp > 921.889)) {
            $material[$loop_index] = "DURAFRSI_blankets";
            $inputfile[$loop_index]="DURAFRSI_TiAl_3inputs.in";
          $score[$loop_index]=4;
         }
         elsif (($maxradeqtemp <= 921.889) and ($maxradeqtemp > 699.667)) {
            $material[$loop_index] = "AFRSI_blankets";
            $inputfile[$loop_index]="AFRSI_TiAl_3inputs.in";
          $score[$loop_index]=3;
         }
         elsif (($maxradeqtemp <= 699.667) and ($maxradeqtemp > 505.222)) {
            $material[$loop_index] = "PBI_blankets";
            $inputfile[$loop_index]="PBI_TiAl_3inputs.in";
          $score[$loop_index]=2;
         }
         elsif (($maxradeqtemp <= 505.222) and ($maxradeqtemp > 0.0)) {
            $material[$loop_index] = "FRSI_blankets";
            $inputfile="FRSI_TiAl_3inputs.in";
          $score[$loop_index]=1;
         }
      }
   }
   elsif ($backfacematerial eq "GrEx") {
      if ($bodyside eq "windward") {
         if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
            $material[$loop_index] = "RCC_tiles";
            $inputfile[$loop_index]="RCC_GrEx_5inputs.in";
          $score[$loop_index]=5;
         }
         elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1699.667)) {
            $material[$loop_index] = "SiC_tiles";
            $inputfile[$loop_index]="SIC_GrEx_5inputs.in";
          $score[$loop_index]=4;
         }
         elsif (($maxradeqtemp <= 1699.667) and ($maxradeqtemp > 1644.111)) {
            $material[$loop_index] = "AETB12_tiles";
            $inputfile[$loop_index]="AETB_12_GrEx_5inputs.in";
          $score[$loop_index]=3;
         }
```

```
        elsif (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 1588.556)) {
           $material[$loop_index] = "AETB8_tiles";
           $inputfile[$loop_index]="AETB_8_GrEx_5inputs.in";
         $score[$loop_index]=2;
        }
        elsif (($maxradeqtemp <= 1588.556) and ($maxradeqtemp > 0.0)) {
           $material[$loop_index] = "LI900_tiles";
           $inputfile[$loop_index]="LI_900_GrEx_5inputs.in";
         $score[$loop_index]=1;
        }
     }
     elsif ($bodyside eq "leeward") {
        if(($maxradeqtemp <= 1477.444) and ($maxradeqtemp > 1366.333)) {
           $material[$loop_index] = "CFBI_blankets";
           $inputfile[$loop_index]="CFBI_GrEx_3inputs.in";
         $score[$loop_index]=7;
        }
        elsif (($maxradeqtemp <= 1366.333) and ($maxradeqtemp > 1310.778)) {
           $material[$loop_index] = "AFRSI2500_blankets";
           $inputfile[$loop_index]="AFRSI2500_GrEx_3inputs.in";
         $score[$loop_index]=6;
        }
        elsif (($maxradeqtemp <= 1310.778) and ($maxradeqtemp > 1255.222)) {
           $material[$loop_index] = "AFRSI2200_blankets";
           $inputfile[$loop_index]="AFRSI2200_GrEx_3inputs.in";
         $score[$loop_index]=5;
        }
        elsif (($maxradeqtemp <= 1255.222) and ($maxradeqtemp > 921.889)) {
           $material[$loop_index] = "DURAFRSI_blankets";
           $inputfile[$loop_index]="DURAFRSI_GrEx_3inputs.in";
         $score[$loop_index]=4;
        }
        elsif (($maxradeqtemp <= 921.889) and ($maxradeqtemp > 699.667)) {
           $material[$loop_index] = "AFRSI_blankets";
           $inputfile[$loop_index]="AFRSI_GrEx_3inputs.in";
         $score[$loop_index]=3;
        }
        elsif (($maxradeqtemp <= 699.667) and ($maxradeqtemp > 505.222)) {
           $material[$loop_index] = "PBI_blankets";
           $inputfile[$loop_index]="PBI_GrEx_3inputs.in";
         $score[$loop_index]=2;
        }
        elsif (($maxradeqtemp <= 505.222) and ($maxradeqtemp > 0.0)) {
           $material[$loop_index] = "FRSI_blankets";
           $inputfile[$loop_index]="FRSI_GrEx_3inputs.in";
         $score[$loop_index]=1;
        }
     }
  }
}
elsif ($tps_family eq "nextgen") {
  if ($backfacematerial eq "TiAl") {
     if ($bodyside eq "windward") {
        if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
           $material[$loop_index] = "RCC_tiles";
           $inputfile[$loop_index]="RCC_TiAl_5inputs.in";
         $score[$loop_index]=3;
        }
        elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1588.556)) {
           $material[$loop_index] = "SiC_tiles";
```

```
          $inputfile[$loop_index]="SIC_TiAl_5inputs.in";
         $score[$loop_index]=2;
        }
        elsif (($maxradeqtemp <= 1588.556) and ($maxradeqtemp > 200.00)) {
          $material[$loop_index] = "LI900_tiles";
          $inputfile[$loop_index]="LI_900_TiAl_5inputs.in";
         $score[$loop_index]=1;
        }
     }
     elsif ($bodyside eq "leeward") {
        if(($maxradeqtemp <= 1477.444) and ($maxradeqtemp > 1366.333)) {
          $material[$loop_index] = "CFBI_blankets";
          $inputfile[$loop_index]="CFBI_TiAl_3inputs.in";
         $score[$loop_index]=4;
        }
        elsif (($maxradeqtemp <= 1366.333) and ($maxradeqtemp > 1310.778)) {
          $material[$loop_index] = "AFRSI2500_blankets";
          $inputfile[$loop_index]="AFRSI2500_TiAl_3inputs.in";
         $score[$loop_index]=3;
        }
        elsif (($maxradeqtemp <= 1310.778) and ($maxradeqtemp > 1255.222)) {
          $material[$loop_index] = "AFRSI2200_blankets";
          $inputfile[$loop_index]="AFRSI2200_TiAl_3inputs.in";
         $score[$loop_index]=2;
        }
        elsif (($maxradeqtemp <= 1255.222) and ($maxradeqtemp > 200.00)) {
          $material[$loop_index] = "DURAFRSI_blankets";
          $inputfile[$loop_index]="DURAFRSI_TiAl_3inputs.in";
         $score[$loop_index]=1;
        }
     }
  }
  elsif ($backfacematerial eq "GrEx") {
     if ($bodyside eq "windward") {
        if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
          $material[$loop_index] = "RCC_tiles";
          $inputfile[$loop_index]="RCC_GrEx_5inputs.in";
         $score[$loop_index]=3;
        }
        elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1588.556)) {
          $material[$loop_index] = "SiC_tiles";
          $inputfile[$loop_index]="SIC_GrEx_5inputs.in";
         $score[$loop_index]=2;
        }
        elsif (($maxradeqtemp <= 1588.556) and ($maxradeqtemp > 200.00)) {
          $material[$loop_index] = "LI900_tiles";
          $inputfile[$loop_index]="LI_900_GrEx_5inputs.in";
         $score[$loop_index]=1;
        }
     }
     elsif ($bodyside eq "leeward") {
        if(($maxradeqtemp <= 1477.444) and ($maxradeqtemp > 1366.333)) {
          $material[$loop_index] = "CFBI_blankets";
          $inputfile[$loop_index]="CFBI_GrEx_3inputs.in";
         $score[$loop_index]=4;
        }
        elsif (($maxradeqtemp <= 1366.333) and ($maxradeqtemp > 1310.778)) {
          $material[$loop_index] = "AFRSI2500_blankets";
          $inputfile[$loop_index]="AFRSI2500_GrEx_3inputs.in";
         $score[$loop_index]=3;
```

```
            }
            elsif (($maxradeqtemp <= 1310.778) and ($maxradeqtemp > 1255.222)) {
               $material[$loop_index] = "AFRSI2200_blankets";
               $inputfile[$loop_index]="AFRSI2200_GrEx_3inputs.in";
             $score[$loop_index]=2;
            }
            elsif (($maxradeqtemp <= 1255.222) and ($maxradeqtemp > 200.00)) {
               $material[$loop_index] = "DURAFRSI_blankets";
               $inputfile[$loop_index]="DURAFRSI_GrEx_3inputs.in";
             $score[$loop_index]=1;
            }
         }
      }
   }
   elsif ($tps_family eq "shuttle") {
      if ($backfacematerial eq "TiAl") {
         if ($bodyside eq "windward") {
            if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
               $material[$loop_index] = "RCC_tiles";
               $inputfile[$loop_index]="RCC_TiAl_5inputs.in";
             $score[$loop_index]=3;
            }
            elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1644.111)) {
               $material[$loop_index] = "LI2200_tiles";
               $inputfile[$loop_index]="LI_2200_TiAl_5inputs.in";
             $score[$loop_index]=2;
            }
            elsif (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 200.0)) {
               $material[$loop_index] = "FRCI_tiles";
               $inputfile[$loop_index]="FRCI_12_TiAl_5inputs.in";
             $score[$loop_index]=1;
            }
         }
         elsif ($bodyside eq "leeward") {
            if (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 921.889)) {
               $material[$loop_index] = "FRCI_tiles";
               $inputfile[$loop_index]="FRCI_12_TiAl_5inputs.in";
             $score[$loop_index]=3;
            }
            if (($maxradeqtemp <= 921.889) and ($maxradeqtemp > 505.222)) {
               $material[$loop_index] = "AFRSI_blankets";
               $inputfile[$loop_index]="AFRSI_TiAl_3inputs.in";
             $score[$loop_index]=2;
            }
            elsif (($maxradeqtemp <= 505.222) and ($maxradeqtemp > 200.0)) {
               $material[$loop_index] = "FRSI_blankets";
               $inputfile="FRSI_TiAl_3inputs.in";
             $score[$loop_index]=1;
            }
         }
      }
      elsif ($backfacematerial eq "GrEx") {
         if ($bodyside eq "windward") {
            if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
               $material[$loop_index] = "RCC_tiles";
               $inputfile[$loop_index]="RCC_GrEx_5inputs.in";
             $score[$loop_index]=3;
            }
            elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1644.111)) {
               $material[$loop_index] = "LI2200_tiles";
```

```
            $inputfile[$loop_index]="LI_2200_GrEx_5inputs.in";
           $score[$loop_index]=2;
          }
          elsif (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 200.0)) {
            $material[$loop_index] = "FRCI_tiles";
            $inputfile[$loop_index]="FRCI_12_GrEx_5inputs.in";
           $score[$loop_index]=1;
          }
       }
       elsif ($bodyside eq "leeward") {
          if (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 921.889)) {
            $material[$loop_index] = "FRCI_tiles";
            $inputfile[$loop_index]="FRCI_12_GrEx_5inputs.in";
           $score[$loop_index]=3;
          }
          if (($maxradeqtemp <= 921.889) and ($maxradeqtemp > 505.222)) {
            $material[$loop_index] = "AFRSI_blankets";
            $inputfile[$loop_index]="AFRSI_GrEx_3inputs.in";
           $score[$loop_index]=2;
          }
          elsif (($maxradeqtemp <= 505.222) and ($maxradeqtemp > 200.0)) {
            $material[$loop_index] = "FRSI_blankets";
            $inputfile="FRSI_GrEx_3inputs.in";
           $score[$loop_index]=1;
          }
       }
    }
}
}

# -----------------------------------------------------------
# logic check to prevent patchwork of materials on RLV
# -----------------------------------------------------------

$RCC_count = 0;
$SIC_count = 0;
$AETB12_count = 0;
$AETB8_count = 0;
$LI900_count = 0;
$LI2200_count=0;
$FRCI_count=0;
$CFBI_count = 0;
$AFRSI2500_count = 0;
$AFRSI2200_count = 0;
$AFRSI_count = 0;
$PBI_count = 0;
$FRSI_count = 0;
$DURAFRSI_count=0;

$k=0;
for ($k = 3; $k <= $end; $k++) {
   if (($material[$k] eq $material[$k-2]) && ($material[$k] ne $material[$k-1])
&& ($score[$k] > $score[$k-1])) {
      $material[$k-1] = $material[$k];
      $inputfile[$k-1]=$inputfile[$k];
   }
   elsif (($material[$k] eq $material[$k-2]) && ($material[$k] ne $material[$k-
1]) && ($score[$k] < $score[$k-1])) {
      $material[$k-2] = $material[$k-1];
      $inputfile[$k-2]=$inputfile[$k-1];
```

```
      $material[$k] = $material[$k-1];
      $inputfile[$k]=$inputfile[$k-1];
   }
# ------------------------------------------------------------------------
# A counter is placed on each one of the materials chosen.  This allows for the
# tps area to body area ratio to be calculated.
# ------------------------------------------------------------------------
}
$k=0;
for ($k = 1; $k <= $end; $k++) {
   if ($material[$k] eq "RCC_tiles") {
      $RCC_count=$RCC_count+1;
   }
   elsif ($material[$k] eq "SiC_tiles") {
      $SIC_count=$SIC_count+1;
   }
   elsif ($material[$k] eq "AETB12_tiles") {
      $AETB12_count=$AETB12_count+1;
   }
   elsif ($material[$k] eq "AETB8_tiles") {
      $AETB8_count=$AETB8_count+1;
   }
   elsif ($material[$k] eq "LI900_tiles") {
      $LI900_count=$LI900_count+1;
   }
   elsif ($material[$k] eq "LI2200_tiles") {
      $LI2200_count=$LI2200_count+1;
   }
   elsif ($material[$k] eq "FRCI_tiles") {
      $FRCI_count=$FRCI_count+1;
   }
   elsif ($material[$k] eq "CFBI_blankets") {
      $CFBI_count=$CFBI_count+1;
   }
   elsif ($material[$k] eq "AFRSI2500_blankets") {
      $AFRSI2500_count=$AFRSI2500_count+1;
   }
   elsif ($material[$k] eq "AFRSI2200_blankets") {
      $AFRSI2200_count=$AFRSI2200_count+1;
   }
   elsif ($material[$k] eq "AFRSI_blankets") {
      $AFRSI_count=$AFRSI_count+1;
   }
   elsif ($material[$k] eq "DURAFRSI_blankets") {
      $DURAFRSI_count=$DURAFRSI_count+1;
   }
   elsif ($material[$k] eq "PBI_blankets") {
      $PBI_count=$PBI_count+1;
   }
   elsif ($material[$k] eq "FRSI_blankets") {
      $FRSI_count=$FRSI_count+1;
   }
}

# ---------------------------------------------------
# conduct heating analysis
# ---------------------------------------------------

open(FINALOUTPUT,">thickness.txt");
print FINALOUTPUT "TPS design results for $bodyside analysis for $filename \n";
```

```
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   system("touch /home/asdl1/kcowart/public_html/miniver.in");
   system("cp /home/asdl1/kcowart/public_html/$name[$loop_index]
/home/asdl1/kcowart/public_html/miniver.in>> error1 2>&1");
   system("touch inputs.in");
   system("cp $inputfile[$loop_index] inputs.in >> error2 2>&1");
   system("touch material_density");
   system("chmod 777 material_density");

   if ($material eq "RCC_tiles") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material[$loop_index] eq "SiC_tiles") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   else {
      system("/home/asdl1/kcowart/public_html/go.exe>junkoutput");# run TCAT
   }

   system("mv fort.90 $name[$loop_index]_temphist");
   system("mv fort.91 $name[$loop_index]_heatratehist");

   open(TCATOUTPUT,"<fort.12"); # obtain thickness of current TPS material
   $_=<TCATOUTPUT>;
   /^\s*(\S*)/;
   $thickness[$loop_index]=$1;
   close(TCATOUTPUT);

   open(DEN,"<material_density"); # obtain the density of current TPS material
   $_=<DEN>;
   /^\s*(\S*)/;
   $tps_density[$loop_index]=$1;
   close(DEN);

# ---------------------------------------------------------
# print the output to file thickness.txt
# ---------------------------------------------------------

   printf FINALOUTPUT "%8s - %15s - TPS thickness = %8.2f
inches\n",$name[$loop_index],$material[$loop_index],$thickness[$loop_index]*(100
./2.54);
}
close(TEMP);

# ----------------------------------------------------------------------
# calculate the unit weight and average thickness
# ----------------------------------------------------------------------

$avg_thickness=0.0;
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   $avg_thickness=$avg_thickness+$thickness[$loop_index];
}
$avg_thickness=$avg_thickness/$end;
$avg_thickness=$avg_thickness*(100./2.54);

$unit_weights=0.;
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
```

```
   $unit_weight=$unit_weight+$tps_density[$loop_index]*$thickness[$loop_index];
}
$unit_weight=($unit_weight/0.4536)/(3.2808**2);
$unit_weight=$unit_weight/$end;

$avg_unit_weight=$unit_weight/$end;

printf FINALOUTPUT "\n";
printf FINALOUTPUT "Average TPS thickness = %8.2f inches\n",$avg_thickness;
printf FINALOUTPUT "Smeared TPS unit_weight = %8.2f lbm/ft^2\n",$unit_weight;

# ------------------------------------------------------------------------
# calculate the area percentages for each of the TPS materials used
# ------------------------------------------------------------------------

if ($tps_family eq "all") {
   if ($bodyside eq "windward") {
      $RCC_ratio=$RCC_count/$end*$tileratio;
      $SIC_ratio=$SIC_count/$end*$tileratio;
      $AETB12_ratio=$AETB12_count/$end*$tileratio;
      $AETB8_ratio=$AETB8_count/$end*$tileratio;
      $LI900_ratio=$LI900_count/$end*$tileratio;
   }
   elsif ($bodyside eq "leeward") {
      $CFBI_ratio=$CFBI_count/$end*$blanketratio;
      $AFRSI2500_ratio=$AFRSI2500_count/$end*$blanketratio;
      $AFRSI2200_ratio=$AFRSI2200_count/$end*$blanketratio;
      $DURAFRSI_ratio=$DURAFRSI_count/$end*$blanketratio;
      $AFRSI_ratio=$AFRSI_count/$end*$blanketratio;
      $PBI_ratio=$PBI_count/$end*$blanketratio;
      $FRSI_ratio=$FRSI_count/$end*$blanketratio;
   }
}
elsif ($tps_family eq "nextgen") {
   if ($bodyside eq "windward") {
      $RCC_ratio=$RCC_count/$end*$tileratio;
      $SIC_ratio=$SIC_count/$end*$tileratio;
      $LI900_ratio=$LI900_count/$end*$tileratio;
   }
   elsif ($bodyside eq "leeward") {
      $CFBI_ratio=$CFBI_count/$end*$blanketratio;
      $AFRSI2500_ratio=$AFRSI2500_count/$end*$blanketratio;
      $AFRSI2200_ratio=$AFRSI2200_count/$end*$blanketratio;
      $DURAFRSI_ratio=$DURAFRSI_count/$end*$blanketratio;
   }
}
elsif ($tps_family eq "shuttle") {
   if ($bodyside eq "windward") {
      $RCC_ratio=$RCC_count/$end*$tileratio;
      $LI2200_ratio=$LI2200_count/$end*$tileratio;
      $FRCI_ratio=$FRCI_count/$end*$tileratio;
   }
   elsif ($bodyside eq "leeward") {
      $FRCI_ratio=$FRCI_count/$end*$blanketratio;
      $AFRSI_ratio=$AFRSI_count/$end*$blanketratio;
      $FRSI_ratio=$FRSI_count/$end*$blanketratio;
   }
}
$RCC_unitweight=0;
$SIC_unitweight=0;
```

```
$AETB12_unitweight=0;
$AETB8_unitweight=0;
$LI900_unitweight=0;
$LI2200_unitweight=0;
$FRCI_unitweight=0;
$CFBI_unitweight=0;
$AFRSI2500_unitweight=0;
$AFRSI2200_unitweight=0;
$AFRSI_unitweight=0;
$DURAFRSI_unitweight=0;
$PBI_unitweight=0;
$FRSI_unitweight=0;
$k=0;
for ($k = 1; $k <= $end; $k++) {
   if ($material[$k] eq "RCC_tiles") {

$RCC_unitweight=$RCC_unitweight+($tps_density[$k]*$thickness[$k]+0.00254*1577.83
47)/(0.00254+$thickness[$k])/$RCC_count;
   }
   elsif ($material[$k] eq "SiC_tiles") {

$SIC_unitweight=$SIC_unitweight+($tps_density[$k]*$thickness[$k]+0.00254*2400.00
)/(0.00254+$thickness[$k])/$SIC_count;
   }
   elsif ($material[$k] eq "AETB12_tiles") {

$AETB12_unitweight=$AETB12_unitweight+$tps_density[$k]*$thickness[$k]/$AETB12_co
unt;
   }
   elsif ($material[$k] eq "AETB8_tiles") {

$AETB8_unitweight=$AETB8_unitweight+$tps_density[$k]*$thickness[$k]/$AETB8_count
;
   }
   elsif ($material[$k] eq "LI900_tiles") {

$LI900_unitweight=$LI900_unitweight+$tps_density[$k]*$thickness[$k]/$LI900_count
;
   }
   elsif ($material[$k] eq "LI2200_tiles") {

$LI2200_unitweight=$LI2200_unitweight+$tps_density[$k]*$thickness[$k]/$LI2200_co
unt;
   }
   elsif ($material[$k] eq "FRCI_tiles") {

$FRCI_unitweight=$FRCI_unitweight+$tps_density[$k]*$thickness[$k]/$FRCI_count;
   }
   elsif ($material[$k] eq "CFBI_blankets") {

$CFBI_unitweight=$CFBI_unitweight+$tps_density[$k]*$thickness[$k]/$CFBI_count;
   }
   elsif ($material[$k] eq "AFRSI2500_blankets") {

$AFRSI2500_unitweight=$AFRSI2500_unitweight+$tps_density[$k]*$thickness[$k]/AFRS
I2500_count;
   }
   elsif ($material[$k] eq "AFRSI2200_blankets") {
```

```
$AFRSI2200_unitweight=$AFRSI2200_unitweight+$tps_density[$k]*$thickness[$k]/$AFR
SI2200_count;
   }
   elsif ($material[$k] eq "AFRSI_blankets") {

$AFRSI_unitweight=$AFRSI_unitweight+$tps_density[$k]*$thickness[$k]/$AFRSI_count
;
   }
   elsif ($material[$k] eq "DURAFRSI_blankets") {

$DURAFRSI_unitweight=$DURAFRSI_unitweight+$tps_density[$k]*$thickness[$k]/$DURAF
RSI_count;
   }
   elsif ($material[$k] eq "PBI_blankets") {

$PBI_unitweight=$PBI_unitweight+$tps_density[$k]*$thickness[$k]/$PBI_count;
   }
   elsif ($material[$k] eq "FRSI_blankets") {

$FRSI_unitweight=$FRSI_unitweight+$tps_density[$k]*$thickness[$k]/$FRSI_count;
   }
}

if (($sharp eq "yes") && ($bodyside eq "windward")) {
   $sharp_weight=4.6e-3*$cowlarea;      #SHARP weight in lbm
   printf FINALOUTPUT "\n";
   printf FINALOUTPUT "Cowl SHARP TPS: %6.2f lbm/ft
\n",$sharp_weight/$cowllength;
}
if (($sharp eq "yes") && ($bodyside eq "leeward")) {
   $sharp_weight=4.6e-3*$cowlarea;      #SHARP weight in lbm
   printf FINALOUTPUT "\n";
   printf FINALOUTPUT "Cowl SHARP TPS: %6.2f lbm/ft
\n",$sharp_weight/$cowllength;
}

if ($RCC_count > 0) {
   printf FINALOUTPUT "RCC unit weight: %6.2f lbm/ft2\n",
$RCC_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "RCC TPS Area to Cowl Area Ratio: %6.4f\n", $RCC_ratio;

}
if ($SIC_count > 0) {
   printf FINALOUTPUT "SiC unit weight: %6.2f lbm/ft2\n",
$SIC_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "SiC TPS Area to Cowl Area Ratio: %6.4f\n", $SIC_ratio;
}
if ($AETB12_count > 0) {
   printf FINALOUTPUT "AETB12 unit weight: %6.2f lbm/ft2\n",
$AETB12_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AETB12 TPS Area to Cowl Area Ratio: %6.4f\n",
$AETB12_ratio;
}
if ($AETB8_count > 0) {
   printf FINALOUTPUT "AETB8 unit weight: %6.2f lbm/ft2\n",
$AETB8_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AETB8 TPS Area to Cowl Area Ratio: %6.4f\n",
$AETB8_ratio;
}
```

```
if ($LI900_count > 0) {
   printf FINALOUTPUT "LI900 unit weight: %6.2f lbm/ft2\n",
$LI900_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "LI900 TPS Area to Cowl Area Ratio: %6.4f\n",
$LI900_ratio;
}
if ($LI2200_count > 0) {
   printf FINALOUTPUT "LI2200 unit weight: %6.2f lbm/ft2\n",
$LI2200_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "LI2200 TPS Area to Cowl Area Ratio: %6.4f\n",
$LI2200_ratio;
}
if ($FRCI_count > 0) {
   printf FINALOUTPUT "FRCI unit weight: %6.2f lbm/ft2\n",
$FRCI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "FRCI TPS Area to Cowl Area Ratio: %6.4f\n", $FRCI_ratio;
}
if ($CFBI_count > 0) {
   printf FINALOUTPUT "CFBI unit weight: %6.2f lbm/ft2\n",
$CFBI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "CFBI TPS Area to Cowl Area Ratio: %6.4f\n", $CFBI_ratio;
}
if ($AFRSI2500_count > 0) {
   printf FINALOUTPUT "AFRSI2500 unit weight: %6.2f lbm/ft2\n",
$AFRSI2500_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AFRSI2500 TPS Area to Cowl Area Ratio: %6.4f\n",
$AFRSI2500_ratio;
}
if ($AFRSI2200_count > 0) {
   printf FINALOUTPUT "AFRSI2200 unit weight: %6.2f lbm/ft2\n",
$AFRSI2200_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AFRSI2200 TPS Area to Cowl Area Ratio: %6.4f\n",
$AFRSI2200_ratio;
}
if ($AFRSI_count > 0) {
   printf FINALOUTPUT "AFRSI unit weight: %6.2f lbm/ft2\n",
$AFRSI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AFRSI TPS Area to Cowl Area Ratio: %6.4f\n",
$AFRSI_ratio;
}
if ($DURAFRSI_count > 0) {
   printf FINALOUTPUT "DURAFRSI unit weight: %6.2f lbm/ft2\n",
$DURAFRSI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "DURAFRSI TPS Area to Cowl Area Ratio: %6.4f\n",
$DURAFRSI_ratio;
}
if ($PBI_count > 0) {
   printf FINALOUTPUT "PBI unit weight: %6.2f lbm/ft2\n",
$PBI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "PBI TPS Area to Body Cowl Ratio: %6.4f\n", $PBI_ratio;
}
if ($FRSI_count > 0) {
   printf FINALOUTPUT "FRSI unit weight: %6.2f lbm/ft2\n",
$FRSI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "FRSI TPS Area to Body Cowl Ratio: %6.4f\n", $FRSI_ratio;
}

close(FINALOUTPUT);
```

```
# ------------------------------------------------------------------
# print desired output to webpage
# ------------------------------------------------------------------

open(OUT,"<thickness.txt");            #opens the input file for TCAT script
open(WEB,"<tcat_output.html");  #open tcat html file
print "Content-type:text/html\n\n";
while(<WEB>){
   if(/Insert stuff here/){
      print "<BR>\n";
      while(<OUT>){
         print "<BR>\n";
        print $_;
      }
   }
   else{
     print $_;
   }
}
close(WEB);
close(OUT);
```

```
#!/usr/sbin/perl

# ---------------------------------------------------
# Obtain Information From Web Browser
# ---------------------------------------------------
#
# This section of code allows a cgi script to run from a web browser.
# The information is passed using the POST and the input information is
# parceled out in a mesh type variable and stores in $FORM{$name}.
#

read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs)
{
       ($name, $value) = split(/=/, $pair);
       $value =~ tr/+/ /;
       $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
       $value =~ s/~!/ ~!/g;
       $FORM{$name} = $value;
}

# ------------------------------------------------------------
# assign information from input website to appropriate variable
# ------------------------------------------------------------

$wingarea=$FORM{'wingarea'};
$tileratio=$FORM{'tileratio'};
$blanketratio=$FORM{'blanketratio'};
$winglength=$FORM{'winglength'};
$bodyside=$FORM{'side'};
$backfacematerial=$FORM{'backface'};
$sharp=$FORM{'sharp'};
$tps_family=$FORM{'tpsfamily'};
$filename=$FORM{'filename'};

# -------------------------------------------------------------------------
# create the body point files
# -------------------------------------------------------------------------

$i=0;                          #initialization of counter index "i"
$filecount=0;
open(FILE,"<$filename");       #open statement with filehandle FILE

until(eof FILE) {              #goes through the MINVER file until the end is
reached
   START: $_=<FILE>;           #skips the first line of the data file

   if (/^.{73}(\S*)/) {               #searches for the bodypoint label
   $title=$1;
   $filecount=$filecount+1;
   } # end of if statement

   while(<FILE>) {                    #while loop that reads thru each line of
the file
      $i=$i+1;

/^\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S*)\s*(\S
*)\s*(\S*)\s*(\S*)\s*(\S*)/;
      $time[$i]=$1;            #array declaration for time
```

```
        $tradeq[$i]=$9;                    #array declaration for rad eq temp
        $qconv[$i]=$10;                    #array declaration for conv heat rate
        $n=$i;
        if ($1==-100) {
           open(NEWFILE,">point$title"); #open statement for new file "newsample"
          $timesteps=$n-1;
           printf NEWFILE "$timesteps \n";#formatted print statement"
          for($i=1; $i<$n; $i++) {
               printf NEWFILE "%f \n",$time[$i];
               printf NEWFILE "%f \n",$qconv[$i];
               printf NEWFILE "%f \n",$tradeq[$i];
           }  #end of for loop

           close(NEWFILE);
           $i=0;
           goto START
        }  #end of if statement

    }  #end of while loop
 }  #end of until loop
close(FILE);

# ------------------------------------------------------------------------
# loop trough the body point files and determine the materials for each
# body point
# ------------------------------------------------------------------------

$start=1;             #start of the file index
$end=($filecount-1)/2;    #end of the file index
#$end=1;

open(FILE,">inputs_for_outputs");
printf FILE "$bodyside\n";
printf FILE "$filename\n";
printf FILE "$end\n";
close(FILE);

# ------------------------------------------
# determine the TPS based on the Rad Eq Temp
# ------------------------------------------

for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   if ($bodyside eq "leeward") {
      $name[$loop_index]="pointl$loop_index"; # stores the bodypoint filename
   }
   if ($bodyside eq "windward") {
      $name[$loop_index]="pointw$loop_index"; # stores the bodypoint filename
   }

# ---------------------------------------------------
# read inforamtion from body point files
# ---------------------------------------------------

   $maxradeqtemp = 0.0;
   open(FILE,"< $name[$loop_index]");
      $k = readline(*FILE);
      chop($k);
      for ($index = 1; $index <= $k; $index++) {
         $time[$index] = readline(*FILE);
         chop($time[$index]);
```

```
        $qconv[$index] = readline(*FILE);
        chop($qconv[$index]);
        $tradeq[$index] = readline(*FILE);
        chop($tradeq[$index]);
     }
   close(FILE);

# ----------------------------------------------------------------------------
-
# dynamically allowcate files for Radiation Equilibrium Temp for each body point
# ----------------------------------------------------------------------------
-

   open(FILE,"> $name[$loop_index]_radeqtemp");
   for ($index=1; $index <= $k; $index++) {
      printf FILE "%10.2f   %10.2f\n",$time[$index],$tradeq[$index];
   }
   close(FILE);

# --------------------------------------------------
# find the maximum value of the rastemperature
# --------------------------------------------------

   $i = 0;
   $maxradeqtemp=$tradeq[1];
   for($i = 2; $i <= $n; $i++) {
      if ($tradeq[$i] > $maxradeqtemp) {
         $maxradeqtemp=$tradeq[$i];
      }
   }

# -------------------------------------------------------------------------
# make material selection based on backface material,  side of vehicle, and
# rad eq temp, and conduct heating analysis at each body point
# -------------------------------------------------------------------------

if ($tps_family eq "all") {
   if ($backfacematerial eq "TiAl") {
      if ($bodyside eq "windward") {
         if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
            $material[$loop_index] = "RCC_tiles";
            $inputfile[$loop_index]="RCC_TiAl_5inputs.in";
           $score[$loop_index]=5;
         }
         elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1699.667)) {
            $material[$loop_index] = "SiC_tiles";
            $inputfile[$loop_index]="SIC_TiAl_5inputs.in";
           $score[$loop_index]=4;
         }
         elsif (($maxradeqtemp <= 1699.667) and ($maxradeqtemp > 1644.111)) {
            $material[$loop_index] = "AETB12_tiles";
            $inputfile[$loop_index]="AETB_12_TiAl_5inputs.in";
           $score[$loop_index]=3;
         }
         elsif (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 1588.556)) {
            $material[$loop_index] = "AETB8_tiles";
            $inputfile[$loop_index]="AETB_8_TiAl_5inputs.in";
           $score[$loop_index]=2;
         }
         elsif (($maxradeqtemp <= 1588.556) and ($maxradeqtemp > 0.0)) {
```

```
          $material[$loop_index] = "LI900_tiles";
          $inputfile[$loop_index]="LI_900_TiAl_5inputs.in";
        $score[$loop_index]=1;
        }
    }
    elsif ($bodyside eq "leeward") {
        if(($maxradeqtemp <= 1477.444) and ($maxradeqtemp > 1366.333)) {
          $material[$loop_index] = "CFBI_blankets";
          $inputfile[$loop_index]="CFBI_TiAl_3inputs.in";
        $score[$loop_index]=7;
        }
        elsif (($maxradeqtemp <= 1366.333) and ($maxradeqtemp > 1310.778)) {
          $material[$loop_index] = "AFRSI2500_blankets";
          $inputfile[$loop_index]="AFRSI2500_TiAl_3inputs.in";
        $score[$loop_index]=6;
        }
        elsif (($maxradeqtemp <= 1310.778) and ($maxradeqtemp > 1255.222)) {
          $material[$loop_index] = "AFRSI2200_blankets";
          $inputfile[$loop_index]="AFRSI2200_TiAl_3inputs.in";
        $score[$loop_index]=5;
        }
        elsif (($maxradeqtemp <= 1255.222) and ($maxradeqtemp > 921.889)) {
          $material[$loop_index] = "DURAFRSI_blankets";
          $inputfile[$loop_index]="DURAFRSI_TiAl_3inputs.in";
        $score[$loop_index]=4;
        }
        elsif (($maxradeqtemp <= 921.889) and ($maxradeqtemp > 699.667)) {
          $material[$loop_index] = "AFRSI_blankets";
          $inputfile[$loop_index]="AFRSI_TiAl_3inputs.in";
        $score[$loop_index]=3;
        }
        elsif (($maxradeqtemp <= 699.667) and ($maxradeqtemp > 505.222)) {
          $material[$loop_index] = "PBI_blankets";
          $inputfile[$loop_index]="PBI_TiAl_3inputs.in";
        $score[$loop_index]=2;
        }
        elsif (($maxradeqtemp <= 505.222) and ($maxradeqtemp > 0.0)) {
          $material[$loop_index] = "FRSI_blankets";
          $inputfile="FRSI_TiAl_3inputs.in";
        $score[$loop_index]=1;
        }
    }
}
elsif ($backfacematerial eq "GrEx") {
    if ($bodyside eq "windward") {
        if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
          $material[$loop_index] = "RCC_tiles";
          $inputfile[$loop_index]="RCC_GrEx_5inputs.in";
        $score[$loop_index]=5;
        }
        elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1699.667)) {
          $material[$loop_index] = "SiC_tiles";
          $inputfile[$loop_index]="SIC_GrEx_5inputs.in";
        $score[$loop_index]=4;
        }
        elsif (($maxradeqtemp <= 1699.667) and ($maxradeqtemp > 1644.111)) {
          $material[$loop_index] = "AETB12_tiles";
          $inputfile[$loop_index]="AETB_12_GrEx_5inputs.in";
        $score[$loop_index]=3;
        }
```

```
            elsif (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 1588.556)) {
               $material[$loop_index] = "AETB8_tiles";
               $inputfile[$loop_index]="AETB_8_GrEx_5inputs.in";
             $score[$loop_index]=2;
            }
            elsif (($maxradeqtemp <= 1588.556) and ($maxradeqtemp > 0.0)) {
               $material[$loop_index] = "LI900_tiles";
               $inputfile[$loop_index]="LI_900_GrEx_5inputs.in";
             $score[$loop_index]=1;
            }
         }
         elsif ($bodyside eq "leeward") {
            if(($maxradeqtemp <= 1477.444) and ($maxradeqtemp > 1366.333)) {
               $material[$loop_index] = "CFBI_blankets";
               $inputfile[$loop_index]="CFBI_GrEx_3inputs.in";
             $score[$loop_index]=7;
            }
            elsif (($maxradeqtemp <= 1366.333) and ($maxradeqtemp > 1310.778)) {
               $material[$loop_index] = "AFRSI2500_blankets";
               $inputfile[$loop_index]="AFRSI2500_GrEx_3inputs.in";
             $score[$loop_index]=6;
            }
            elsif (($maxradeqtemp <= 1310.778) and ($maxradeqtemp > 1255.222)) {
               $material[$loop_index] = "AFRSI2200_blankets";
               $inputfile[$loop_index]="AFRSI2200_GrEx_3inputs.in";
             $score[$loop_index]=5;
            }
            elsif (($maxradeqtemp <= 1255.222) and ($maxradeqtemp > 921.889)) {
               $material[$loop_index] = "DURAFRSI_blankets";
               $inputfile[$loop_index]="DURAFRSI_GrEx_3inputs.in";
             $score[$loop_index]=4;
            }
            elsif (($maxradeqtemp <= 921.889) and ($maxradeqtemp > 699.667)) {
               $material[$loop_index] = "AFRSI_blankets";
               $inputfile[$loop_index]="AFRSI_GrEx_3inputs.in";
             $score[$loop_index]=3;
            }
            elsif (($maxradeqtemp <= 699.667) and ($maxradeqtemp > 505.222)) {
               $material[$loop_index] = "PBI_blankets";
               $inputfile[$loop_index]="PBI_GrEx_3inputs.in";
             $score[$loop_index]=2;
            }
            elsif (($maxradeqtemp <= 505.222) and ($maxradeqtemp > 0.0)) {
               $material[$loop_index] = "FRSI_blankets";
               $inputfile[$loop_index]="FRSI_GrEx_3inputs.in";
             $score[$loop_index]=1;
            }
         }
      }
   }
   elsif ($tps_family eq "nextgen") {
      if ($backfacematerial eq "TiAl") {
         if ($bodyside eq "windward") {
            if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
               $material[$loop_index] = "RCC_tiles";
               $inputfile[$loop_index]="RCC_TiAl_5inputs.in";
             $score[$loop_index]=3;
            }
            elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1588.556)) {
               $material[$loop_index] = "SiC_tiles";
```

```
        $inputfile[$loop_index]="SIC_TiAl_5inputs.in";
       $score[$loop_index]=2;
      }
      elsif (($maxradeqtemp <= 1588.556) and ($maxradeqtemp > 200.00)) {
        $material[$loop_index] = "LI900_tiles";
        $inputfile[$loop_index]="LI_900_TiAl_5inputs.in";
       $score[$loop_index]=1;
      }
    }
    elsif ($bodyside eq "leeward") {
      if(($maxradeqtemp <= 1477.444) and ($maxradeqtemp > 1366.333)) {
        $material[$loop_index] = "CFBI_blankets";
        $inputfile[$loop_index]="CFBI_TiAl_3inputs.in";
       $score[$loop_index]=4;
      }
      elsif (($maxradeqtemp <= 1366.333) and ($maxradeqtemp > 1310.778)) {
        $material[$loop_index] = "AFRSI2500_blankets";
        $inputfile[$loop_index]="AFRSI2500_TiAl_3inputs.in";
       $score[$loop_index]=3;
      }
      elsif (($maxradeqtemp <= 1310.778) and ($maxradeqtemp > 1255.222)) {
        $material[$loop_index] = "AFRSI2200_blankets";
        $inputfile[$loop_index]="AFRSI2200_TiAl_3inputs.in";
       $score[$loop_index]=2;
      }
      elsif (($maxradeqtemp <= 1255.222) and ($maxradeqtemp > 200.00)) {
        $material[$loop_index] = "DURAFRSI_blankets";
        $inputfile[$loop_index]="DURAFRSI_TiAl_3inputs.in";
       $score[$loop_index]=1;
      }
    }
  }
  elsif ($backfacematerial eq "GrEx") {
    if ($bodyside eq "windward") {
      if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
        $material[$loop_index] = "RCC_tiles";
        $inputfile[$loop_index]="RCC_GrEx_5inputs.in";
       $score[$loop_index]=3;
      }
      elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1588.556)) {
        $material[$loop_index] = "SiC_tiles";
        $inputfile[$loop_index]="SIC_GrEx_5inputs.in";
       $score[$loop_index]=2;
      }
      elsif (($maxradeqtemp <= 1588.556) and ($maxradeqtemp > 200.00)) {
        $material[$loop_index] = "LI900_tiles";
        $inputfile[$loop_index]="LI_900_GrEx_5inputs.in";
       $score[$loop_index]=1;
      }
    }
    elsif ($bodyside eq "leeward") {
      if(($maxradeqtemp <= 1477.444) and ($maxradeqtemp > 1366.333)) {
        $material[$loop_index] = "CFBI_blankets";
        $inputfile[$loop_index]="CFBI_GrEx_3inputs.in";
       $score[$loop_index]=4;
      }
      elsif (($maxradeqtemp <= 1366.333) and ($maxradeqtemp > 1310.778)) {
        $material[$loop_index] = "AFRSI2500_blankets";
        $inputfile[$loop_index]="AFRSI2500_GrEx_3inputs.in";
       $score[$loop_index]=3;
```

```
            }
            elsif (($maxradeqtemp <= 1310.778) and ($maxradeqtemp > 1255.222)) {
               $material[$loop_index] = "AFRSI2200_blankets";
               $inputfile[$loop_index]="AFRSI2200_GrEx_3inputs.in";
              $score[$loop_index]=2;
            }
            elsif (($maxradeqtemp <= 1255.222) and ($maxradeqtemp > 200.00)) {
               $material[$loop_index] = "DURAFRSI_blankets";
               $inputfile[$loop_index]="DURAFRSI_GrEx_3inputs.in";
              $score[$loop_index]=1;
            }
         }
      }
   }
   elsif ($tps_family eq "shuttle") {
      if ($backfacematerial eq "TiAl") {
         if ($bodyside eq "windward") {
            if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
               $material[$loop_index] = "RCC_tiles";
               $inputfile[$loop_index]="RCC_TiAl_5inputs.in";
              $score[$loop_index]=3;
            }
            elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1644.111)) {
               $material[$loop_index] = "LI2200_tiles";
               $inputfile[$loop_index]="LI_2200_TiAl_5inputs.in";
              $score[$loop_index]=2;
            }
            elsif (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 200.0)) {
               $material[$loop_index] = "FRCI_tiles";
               $inputfile[$loop_index]="FRCI_12_TiAl_5inputs.in";
              $score[$loop_index]=1;
            }
         }
         elsif ($bodyside eq "leeward") {
            if (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 921.889)) {
               $material[$loop_index] = "FRCI_tiles";
               $inputfile[$loop_index]="FRCI_12_TiAl_5inputs.in";
              $score[$loop_index]=3;
            }
            if (($maxradeqtemp <= 921.889) and ($maxradeqtemp > 505.222)) {
               $material[$loop_index] = "AFRSI_blankets";
               $inputfile[$loop_index]="AFRSI_TiAl_3inputs.in";
              $score[$loop_index]=2;
            }
            elsif (($maxradeqtemp <= 505.222) and ($maxradeqtemp > 200.0)) {
               $material[$loop_index] = "FRSI_blankets";
               $inputfile="FRSI_TiAl_3inputs.in";
              $score[$loop_index]=1;
            }
         }
      }
      elsif ($backfacematerial eq "GrEx") {
         if ($bodyside eq "windward") {
            if(($maxradeqtemp <= 1921.889) and ($maxradeqtemp > 1866.333)) {
               $material[$loop_index] = "RCC_tiles";
               $inputfile[$loop_index]="RCC_GrEx_5inputs.in";
              $score[$loop_index]=3;
            }
            elsif (($maxradeqtemp <= 1866.333) and ($maxradeqtemp > 1644.111)) {
               $material[$loop_index] = "LI2200_tiles";
```

```
            $inputfile[$loop_index]="LI_2200_GrEx_5inputs.in";
           $score[$loop_index]=2;
          }
          elsif (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 200.0)) {
             $material[$loop_index] = "FRCI_tiles";
             $inputfile[$loop_index]="FRCI_12_GrEx_5inputs.in";
           $score[$loop_index]=1;
          }
       }
       elsif ($bodyside eq "leeward") {
          if (($maxradeqtemp <= 1644.111) and ($maxradeqtemp > 921.889)) {
             $material[$loop_index] = "FRCI_tiles";
             $inputfile[$loop_index]="FRCI_12_GrEx_5inputs.in";
           $score[$loop_index]=3;
          }
          if (($maxradeqtemp <= 921.889) and ($maxradeqtemp > 505.222)) {
             $material[$loop_index] = "AFRSI_blankets";
             $inputfile[$loop_index]="AFRSI_GrEx_3inputs.in";
           $score[$loop_index]=2;
          }
          elsif (($maxradeqtemp <= 505.222) and ($maxradeqtemp > 200.0)) {
             $material[$loop_index] = "FRSI_blankets";
             $inputfile="FRSI_GrEx_3inputs.in";
           $score[$loop_index]=1;
          }
       }
    }
}
}

# -----------------------------------------------------------
# logic check to prevent patchwork of materials on RLV
# -----------------------------------------------------------

$RCC_count = 0;
$SIC_count = 0;
$AETB12_count = 0;
$AETB8_count = 0;
$LI900_count = 0;
$LI2200_count=0;
$FRCI_count=0;
$CFBI_count = 0;
$AFRSI2500_count = 0;
$AFRSI2200_count = 0;
$AFRSI_count = 0;
$PBI_count = 0;
$FRSI_count = 0;
$DURAFRSI_count=0;

$k=0;
for ($k = 3; $k <= $end; $k++) {
   if (($material[$k] eq $material[$k-2]) && ($material[$k] ne $material[$k-1])
&& ($score[$k] > $score[$k-1])) {
      $material[$k-1] = $material[$k];
      $inputfile[$k-1]=$inputfile[$k];
   }
   elsif (($material[$k] eq $material[$k-2]) && ($material[$k] ne $material[$k-
1]) && ($score[$k] < $score[$k-1])) {
      $material[$k-2] = $material[$k-1];
      $inputfile[$k-2]=$inputfile[$k-1];
```

```
      $material[$k] = $material[$k-1];
      $inputfile[$k]=$inputfile[$k-1];
   }
# -----------------------------------------------------------------------------
# A counter is placed on each one of the materials chosen.  This allows for the
# tps area to body area ratio to be calculated.
# -----------------------------------------------------------------------------
}
$k=0;
for ($k = 1; $k <= $end; $k++) {
   if ($material[$k] eq "RCC_tiles") {
      $RCC_count=$RCC_count+1;
   }
   elsif ($material[$k] eq "SiC_tiles") {
      $SIC_count=$SIC_count+1;
   }
   elsif ($material[$k] eq "AETB12_tiles") {
      $AETB12_count=$AETB12_count+1;
   }
   elsif ($material[$k] eq "AETB8_tiles") {
      $AETB8_count=$AETB8_count+1;
   }
   elsif ($material[$k] eq "LI900_tiles") {
      $LI900_count=$LI900_count+1;
   }
   elsif ($material[$k] eq "LI2200_tiles") {
      $LI2200_count=$LI2200_count+1;
   }
   elsif ($material[$k] eq "FRCI_tiles") {
      $FRCI_count=$FRCI_count+1;
   }
   elsif ($material[$k] eq "CFBI_blankets") {
      $CFBI_count=$CFBI_count+1;
   }
   elsif ($material[$k] eq "AFRSI2500_blankets") {
      $AFRSI2500_count=$AFRSI2500_count+1;
   }
   elsif ($material[$k] eq "AFRSI2200_blankets") {
      $AFRSI2200_count=$AFRSI2200_count+1;
   }
   elsif ($material[$k] eq "AFRSI_blankets") {
      $AFRSI_count=$AFRSI_count+1;
   }
   elsif ($material[$k] eq "DURAFRSI_blankets") {
      $DURAFRSI_count=$DURAFRSI_count+1;
   }
   elsif ($material[$k] eq "PBI_blankets") {
      $PBI_count=$PBI_count+1;
   }
   elsif ($material[$k] eq "FRSI_blankets") {
      $FRSI_count=$FRSI_count+1;
   }
}

# --------------------------------------------------
# conduct heating analysis
# --------------------------------------------------

open(FINALOUTPUT,">thickness.txt");
print FINALOUTPUT "TPS design results for $bodyside analysis for $filename \n";
```

```
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   system("touch /home/asdl1/kcowart/public_html/miniver.in");
   system("cp /home/asdl1/kcowart/public_html/$name[$loop_index]
/home/asdl1/kcowart/public_html/miniver.in>> error1 2>&1");
   system("touch inputs.in");
   system("cp $inputfile[$loop_index] inputs.in >> error2 2>&1");
   system("touch material_density");
   system("chmod 777 material_density");

   if ($material[$loop_index] eq "RCC_tiles") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   elsif ($material[$loop_index] eq "SiC_tiles") {
      system("/home/asdl1/kcowart/public_html/go2.exe>junkoutput");     # run
TCAT
   }
   else {
      system("/home/asdl1/kcowart/public_html/go.exe>junkoutput");# run TCAT
   }

   system("mv fort.90 $name[$loop_index]_temphist");
   system("mv fort.91 $name[$loop_index]_heatratehist");

   open(TCATOUTPUT,"<fort.12"); # obtain thickness of current TPS material
   $_=<TCATOUTPUT>;
   /^\s*(\S*)/;
   $thickness[$loop_index]=$1;
   close(TCATOUTPUT);

   open(DEN,"<material_density"); # obtain the density of current TPS material
   $_=<DEN>;
   /^\s*(\S*)/;
   $tps_density[$loop_index]=$1;
   close(DEN);

# --------------------------------------------------------
# print the output to file thickness.txt
# --------------------------------------------------------

   printf FINALOUTPUT "%8s - %15s - TPS thickness = %8.2f
inches\n",$name[$loop_index],$material[$loop_index],$thickness[$loop_index]*(100
./2.54);
}
close(TEMP);

# ------------------------------------------------------------------------
# calculate the unit weight and average thickness
# ------------------------------------------------------------------------

$avg_thickness=0.0;
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
   $avg_thickness=$avg_thickness+$thickness[$loop_index];
}
$avg_thickness=$avg_thickness/$end;
$avg_thickness=$avg_thickness*(100./2.54);

$unit_weights=0.;
for ($loop_index=$start; $loop_index <= $end; $loop_index++) {
```

```
   $unit_weight=$unit_weight+$tps_density[$loop_index]*$thickness[$loop_index];
}
$unit_weight=($unit_weight/0.4536)/(3.2808**2);
$unit_weight=$unit_weight/$end;

$avg_unit_weight=$unit_weight/$end;

printf FINALOUTPUT "\n";
printf FINALOUTPUT "Average TPS thickness = %8.2f inches\n",$avg_thickness;
printf FINALOUTPUT "Smeared TPS unit_weight = %8.2f lbm/ft^2\n",$unit_weight;

# ------------------------------------------------------------------
# calculate the area percentages for each of the TPS materials used
# ------------------------------------------------------------------

if ($tps_family eq "all") {
   if ($bodyside eq "windward") {
      $RCC_ratio=$RCC_count/$end*$tileratio;
      $SIC_ratio=$SIC_count/$end*$tileratio;
      $AETB12_ratio=$AETB12_count/$end*$tileratio;
      $AETB8_ratio=$AETB8_count/$end*$tileratio;
      $LI900_ratio=$LI900_count/$end*$tileratio;
   }
   elsif ($bodyside eq "leeward") {
      $CFBI_ratio=$CFBI_count/$end*$blanketratio;
      $AFRSI2500_ratio=$AFRSI2500_count/$end*$blanketratio;
      $AFRSI2200_ratio=$AFRSI2200_count/$end*$blanketratio;
      $DURAFRSI_ratio=$DURAFRSI_count/$end*$blanketratio;
      $AFRSI_ratio=$AFRSI_count/$end*$blanketratio;
      $PBI_ratio=$PBI_count/$end*$blanketratio;
      $FRSI_ratio=$FRSI_count/$end*$blanketratio;
   }
}
elsif ($tps_family eq "nextgen") {
   if ($bodyside eq "windward") {
      $RCC_ratio=$RCC_count/$end*$tileratio;
      $SIC_ratio=$SIC_count/$end*$tileratio;
      $LI900_ratio=$LI900_count/$end*$tileratio;
   }
   elsif ($bodyside eq "leeward") {
      $CFBI_ratio=$CFBI_count/$end*$blanketratio;
      $AFRSI2500_ratio=$AFRSI2500_count/$end*$blanketratio;
      $AFRSI2200_ratio=$AFRSI2200_count/$end*$blanketratio;
      $DURAFRSI_ratio=$DURAFRSI_count/$end*$blanketratio;
   }
}
elsif ($tps_family eq "shuttle") {
   if ($bodyside eq "windward") {
      $RCC_ratio=$RCC_count/$end*$tileratio;
      $LI2200_ratio=$LI2200_count/$end*$tileratio;
      $FRCI_ratio=$FRCI_count/$end*$tileratio;
   }
   elsif ($bodyside eq "leeward") {
      $FRCI_ratio=$FRCI_count/$end*$blanketratio;
      $AFRSI_ratio=$AFRSI_count/$end*$blanketratio;
      $FRSI_ratio=$FRSI_count/$end*$blanketratio;
   }
}
$RCC_unitweight=0;
$SIC_unitweight=0;
```

```
$AETB12_unitweight=0;
$AETB8_unitweight=0;
$LI900_unitweight=0;
$LI2200_unitweight=0;
$FRCI_unitweight=0;
$CFBI_unitweight=0;
$AFRSI2500_unitweight=0;
$AFRSI2200_unitweight=0;
$AFRSI_unitweight=0;
$DURAFRSI_unitweight=0;
$PBI_unitweight=0;
$FRSI_unitweight=0;
$k=0;
for ($k = 1; $k <= $end; $k++) {
   if ($material[$k] eq "RCC_tiles") {

$RCC_unitweight=$RCC_unitweight+($tps_density[$k]*$thickness[$k]+0.00254*1577.83
47)/(0.00254+$thickness[$k])/$RCC_count;
   }
   elsif ($material[$k] eq "SiC_tiles") {

$SIC_unitweight=$SIC_unitweight+($tps_density[$k]*$thickness[$k]+0.00254*2400.00
)/(0.00254+$thickness[$k])/$SIC_count;
   }
   elsif ($material[$k] eq "AETB12_tiles") {

$AETB12_unitweight=$AETB12_unitweight+$tps_density[$k]*$thickness[$k]/$AETB12_co
unt;
   }
   elsif ($material[$k] eq "AETB8_tiles") {

$AETB8_unitweight=$AETB8_unitweight+$tps_density[$k]*$thickness[$k]/$AETB8_count
;
   }
   elsif ($material[$k] eq "LI900_tiles") {

$LI900_unitweight=$LI900_unitweight+$tps_density[$k]*$thickness[$k]/$LI900_count
;
   }
   elsif ($material[$k] eq "LI2200_tiles") {

$LI2200_unitweight=$LI2200_unitweight+$tps_density[$k]*$thickness[$k]/$LI2200_co
unt;
   }
   elsif ($material[$k] eq "FRCI_tiles") {

$FRCI_unitweight=$FRCI_unitweight+$tps_density[$k]*$thickness[$k]/$FRCI_count;
   }
   elsif ($material[$k] eq "CFBI_blankets") {

$CFBI_unitweight=$CFBI_unitweight+$tps_density[$k]*$thickness[$k]/$CFBI_count;
   }
   elsif ($material[$k] eq "AFRSI2500_blankets") {

$AFRSI2500_unitweight=$AFRSI2500_unitweight+$tps_density[$k]*$thickness[$k]/AFRS
I2500_count;
   }
   elsif ($material[$k] eq "AFRSI2200_blankets") {
```

```
$AFRSI2200_unitweight=$AFRSI2200_unitweight+$tps_density[$k]*$thickness[$k]/$AFR
SI2200_count;
   }
   elsif ($material[$k] eq "AFRSI_blankets") {

$AFRSI_unitweight=$AFRSI_unitweight+$tps_density[$k]*$thickness[$k]/$AFRSI_count
;
   }
   elsif ($material[$k] eq "DURAFRSI_blankets") {

$DURAFRSI_unitweight=$DURAFRSI_unitweight+$tps_density[$k]*$thickness[$k]/$DURAF
RSI_count;
   }
   elsif ($material[$k] eq "PBI_blankets") {

$PBI_unitweight=$PBI_unitweight+$tps_density[$k]*$thickness[$k]/$PBI_count;
   }
   elsif ($material[$k] eq "FRSI_blankets") {

$FRSI_unitweight=$FRSI_unitweight+$tps_density[$k]*$thickness[$k]/$FRSI_count;
   }
}

if (($sharp eq "yes") && ($bodyside eq "windward")) {
   $sharp_weight=4.6e-3*$wingarea;      #SHARP weight in lbm
   $rcc_unit_weight=32.83;      #RCC unit weight
   $rcc_area_to_body_area=0.0075;

$tile_unit_weight=$avg_thickness*3.2808*$tps_density*(1/0.4536)*(1/3.2808)**3;
       #obtain density of tps
   $tile_area_to_body_area=$bodytile;
   printf FINALOUTPUT "\n";
   printf FINALOUTPUT "Wing SHARP TPS: %6.2f lbm/ft
\n",$sharp_weight/$winglength;
}
if (($sharp eq "yes") && ($bodyside eq "leeward")) {
   $sharp_weight=4.6e-3*$wingarea;      #SHARP weight in lbm
   printf FINALOUTPUT "\n";
   printf FINALOUTPUT "Wing SHARP TPS: %6.2f lbm/ft
\n",$sharp_weight/$winglength;
}

if ($RCC_count > 0) {
   printf FINALOUTPUT "RCC unit weight: %6.2f lbm/ft2\n",
$RCC_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "RCC TPS Area to Wing Area Ratio: %6.4f\n", $RCC_ratio;

}
if ($SIC_count > 0) {
   printf FINALOUTPUT "SiC unit weight: %6.2f lbm/ft2\n",
$SIC_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "SiC TPS Area to Wing Area Ratio: %6.4f\n", $SIC_ratio;
}
if ($AETB12_count > 0) {
   printf FINALOUTPUT "AETB12 unit weight: %6.2f lbm/ft2\n",
$AETB12_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AETB12 TPS Area to Wing Area Ratio: %6.4f\n",
$AETB12_ratio;
}
```

```
if ($AETB8_count > 0) {
   printf FINALOUTPUT "AETB8 unit weight: %6.2f lbm/ft2\n",
$AETB8_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AETB8 TPS Area to Wing Area Ratio: %6.4f\n",
$AETB8_ratio;
}
if ($LI900_count > 0) {
   printf FINALOUTPUT "LI900 unit weight: %6.2f lbm/ft2\n",
$LI900_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "LI900 TPS Area to Wing Area Ratio: %6.4f\n",
$LI900_ratio;
}
if ($LI2200_count > 0) {
   printf FINALOUTPUT "LI2200 unit weight: %6.2f lbm/ft2\n",
$LI2200_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "LI2200 TPS Area to Wing Area Ratio: %6.4f\n",
$LI2200_ratio;
}
if ($FRCI_count > 0) {
   printf FINALOUTPUT "FRCI unit weight: %6.2f lbm/ft2\n",
$FRCI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "FRCI TPS Area to Wing Area Ratio: %6.4f\n", $FRCI_ratio;
}
if ($CFBI_count > 0) {
   printf FINALOUTPUT "CFBI unit weight: %6.2f lbm/ft2\n",
$CFBI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "CFBI TPS Area to Wing Area Ratio: %6.4f\n", $CFBI_ratio;
}
if ($AFRSI2500_count > 0) {
   printf FINALOUTPUT "AFRSI2500 unit weight: %6.2f lbm/ft2\n",
$AFRSI2500_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AFRSI2500 TPS Area to Wing Area Ratio: %6.4f\n",
$AFRSI2500_ratio;
}
if ($AFRSI2200_count > 0) {
   printf FINALOUTPUT "AFRSI2200 unit weight: %6.2f lbm/ft2\n",
$AFRSI2200_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AFRSI2200 TPS Area to Wing Area Ratio: %6.4f\n",
$AFRSI2200_ratio;
}
if ($AFRSI_count > 0) {
   printf FINALOUTPUT "AFRSI unit weight: %6.2f lbm/ft2\n",
$AFRSI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "AFRSI TPS Area to Wing Area Ratio: %6.4f\n",
$AFRSI_ratio;
}
if ($DURAFRSI_count > 0) {
   printf FINALOUTPUT "DURAFRSI unit weight: %6.2f lbm/ft2\n",
$DURAFRSI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "DURAFRSI TPS Area to Wing Area Ratio: %6.4f\n",
$DURAFRSI_ratio;
}
if ($PBI_count > 0) {
   printf FINALOUTPUT "PBI unit weight: %6.2f lbm/ft2\n",
$PBI_unitweight/(0.4536*3.2808**2);
   printf FINALOUTPUT "PBI TPS Area to Body Wing Ratio: %6.4f\n", $PBI_ratio;
}
if ($FRSI_count > 0) {
   printf FINALOUTPUT "FRSI unit weight: %6.2f lbm/ft2\n",
$FRSI_unitweight/(0.4536*3.2808**2);
```

```
   printf FINALOUTPUT "FRSI TPS Area to Body Wing Ratio: %6.4f\n", $FRSI_ratio;
}

close(FINALOUTPUT);


# -------------------------------------------------------------------
# print desired output to webpage
# -------------------------------------------------------------------

open(OUT,"<thickness.txt");              #opens the input file for TCAT script
open(WEB,"<tcat_output.html");   #open tcat html file
print "Content-type:text/html\n\n";
while(<WEB>){
   if(/Insert stuff here/){
      print "<BR>\n";
      while(<OUT>){
         print "<BR>\n";
        print $_;
      }
   }
   else{
     print $_;
   }
}
close(WEB);
close(OUT);
```

APPENDIX E

TCAT SINGLE TPS DESGIN HTML FILES

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Translational//EN"><HTML>
<HEAD>

  <TITLE> TCAT-bodycalcs-inputs </TITLE>

</HEAD>
<BODY>
  <P> You have selected the body TPS design with several TPS materials option.
     Please fill in the fields below. </P>
  <HR>
  <FORM METHOD=post ACTION="tcat_bodycalc3.cgi" TARGET="TCAT_outputs">

    <P> Enter the approximate wetted body area (ft^2).
    <BR>
    <INPUT TYPE="text" NAME="bodyarea" SIZE="10" MAXLENGTH="10">

    <P> What fraction of body area is covered with tiles?
    <BR>
    <INPUT TYPE="text" NAME="tileratio" SIZE="10" MAXLENGTH="10">

    <P> What fraction of body area is covered with blankets?
    <BR>
    <INPUT TYPE="text" NAME="blanketratio" SIZE="10" MAXLENGTH="10"> </P>

    <! Radio buttons for selecting side of vehicle to analyze>
    <P> Select backface material:
    <BR> <INPUT TYPE="radio" NAME="backface" VALUE="GrEx" CHECKED>Graphite Epoxy
       <INPUT TYPE="radio" NAME="backface" VALUE="TiAl">Titanium Aluminuminide

    <! Radio buttons for selecting side of vehicle to analyze>
    <P> Select side of vehicle for analysis:
    <BR> <INPUT TYPE="radio" NAME="side" VALUE="windward" CHECKED>Windward
       <INPUT TYPE="radio" NAME="side" VALUE="leeward">Leeward

    <P> Are SHARP materials used on the nose?
    <BR> <INPUT TYPE="radio" NAME="sharp" VALUE="yes" CHECKED>Yes
       <INPUT TYPE="radio" NAME="sharp" VALUE="no">No

    <! Radio buttons for selecting family of TPS materials>
    <P> Select TPS material family for analysis:
    <BR> <INPUT TYPE="radio" NAME="tpsfamily" VALUE="all" CHECKED>All Materials in TCAT
       <INPUT TYPE="radio" NAME="tpsfamily" VALUE="nextgen">Next Gen RLV Materials
       <INPUT TYPE="radio" NAME="tpsfamily" VALUE="shuttle">Shuttle Gen RLV Materials

    <BR>
    <A HREF="http://asm.arc.nasa.gov/cgi-bin/tpsx/unrestrict/V2/tpsx-frame.pl" TARGET=new>
    TPSX Material Properties Data Base </A>

    <! Textbox for entering the name of MINIVER file >
    <P> Enter name of MINIVER output file:
    <BR> <INPUT TYPE="text" NAME="filename" SIZE="30"
        MAXLENGTH="30">
    <BR>
    <BR>
    <! Create the submit button>
    <INPUT NAME="Submit" TYPE=submit VALUE="Run TCAT">
    <INPUT NAME="Reset"  TYPE=reset VALUE="Reset Inputs">
    <BR>
    <BR>
```

```
    <A HREF="tcat_selection.html">Return to selection page</A>
  <HR>
  </FORM>
</BODY>
</HTML>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Translational//EN"><HTML>
<HEAD>

  <TITLE> TCAT-bodycalcs-inputs </TITLE>

</HEAD>
<BODY>
  <P> You have selected the body TPS design with several TPS materials option.
    Please fill in the fields below. </P>
  <HR>
  <FORM METHOD=post ACTION="tcat_wingcalc3.cgi" TARGET="TCAT_outputs">

    <P> Enter the approximate wetted wing area (ft^2).
    <BR>
    <INPUT TYPE="text" NAME="wingarea" SIZE="10" MAXLENGTH="10">

    <P> What fraction of wing area is covered with tiles?
    <BR>
    <INPUT TYPE="text" NAME="tileratio" SIZE="10" MAXLENGTH="10">

    <P> What fraction of wing area is covered with blankets?
    <BR>
    <INPUT TYPE="text" NAME="blanketratio" SIZE="10" MAXLENGTH="10"> </P>

    <P> Select backface material:
    <BR> <INPUT TYPE="radio" NAME="backface" VALUE="GrEx" CHECKED>Graphite Epoxy
       <INPUT TYPE="radio" NAME="backface" VALUE="TiAl">Titanium Aluminuminide

    <! Radio buttons for selecting side of wing to analyze>
    <P> Select side of wing for analysis:
    <BR> <INPUT TYPE="radio" NAME="side" VALUE="windward" CHECKED>Windward
       <INPUT TYPE="radio" NAME="side" VALUE="leeward">Leeward

    <P> Are SHARP materials used on the wing leading edges?
    <BR> <INPUT TYPE="radio" NAME="sharp" VALUE="yes" CHECKED>Yes
       <INPUT TYPE="radio" NAME="sharp" VALUE="no">No

    <P> If yes, enter the exposed wing leading edge length (ft).
    <BR>
    <INPUT TYPE="text" NAME="winglength" SIZE="10" MAXLENGTH="10"> </P>

    <! Radio buttons for selecting family of TPS materials>
    <P> Select TPS material family for analysis:
    <BR> <INPUT TYPE="radio" NAME="tpsfamily" VALUE="all" CHECKED>All Materials in TCAT
       <INPUT TYPE="radio" NAME="tpsfamily" VALUE="nextgen">Next Gen RLV Materials
       <INPUT TYPE="radio" NAME="tpsfamily" VALUE="shuttle">Shuttle Gen RLV Materials

    <BR>
    <A HREF="http://asm.arc.nasa.gov/cgi-bin/tpsx/unrestrict/V2/tpsx-frame.pl" TARGET=new>
```

```
    TPSX Material Properties Data Base </A>

    <! Textbox for entering the name of MINIVER file >
    <P> Enter name of MINIVER output file:
    <BR> <INPUT TYPE="text" NAME="filename" SIZE="30"
        MAXLENGTH="30">
    <BR>
    <BR>
    <! Create the submit button>
    <INPUT NAME="Submit" TYPE=submit VALUE="Run TCAT">
    <INPUT NAME="Reset"  TYPE=reset VALUE="Reset Inputs">
    <BR>
    <BR>
    <A HREF="tcat_selection.html">Return to selection page</A>
  <HR>
  </FORM>
</BODY>
</HTML>



<!DOCTYPE HTML PUBLIC "-//W3C DTD HTML 4.0 Translational//EN"><HTML>
<HEAD>

  <TITLE> TCAT-cowlcalcs2-inputs </TITLE>

</HEAD>
<BODY>
  <P> You have selected the cowl TPS design with several TPS materials option.
     Please fill in the fields below. </P>
  <HR>
  <FORM METHOD=post ACTION="tcat_cowlcalc3.cgi" TARGET="TCAT_outputs">

    <P> Enter the approximate wetted cowl area (ft^2).
    <BR>
    <INPUT TYPE="text" NAME="cowlarea" SIZE="10" MAXLENGTH="10">

    <P> What fraction of cowl area is covered with tiles?
    <BR>
    <INPUT TYPE="text" NAME="tileratio" SIZE="10" MAXLENGTH="10">

    <P> What fraction of cowl area is covered with blankets?
    <BR>
    <INPUT TYPE="text" NAME="blanketratio" SIZE="10" MAXLENGTH="10"> </P>

    <P> Enter the cumulative length of cowl and strack edges (ft).
    <BR>
    <INPUT TYPE="text" NAME="cowllength" SIZE="10" MAXLENGTH="10"> </P>

    <P> Select backface material:
    <BR> <INPUT TYPE="radio" NAME="backface" VALUE="GrEx" CHECKED>Graphite Epoxy
       <INPUT TYPE="radio" NAME="backface" VALUE="TiAl">Titanium Aluminuminide

    <! Radio buttons for selecting side of vehicle to analyze>
    <P> Select side of cowl for analysis:
    <BR> <INPUT TYPE="radio" NAME="side" VALUE="windward" CHECKED>Windward
       <INPUT TYPE="radio" NAME="side" VALUE="leeward">Leeward
```

```
<P> Are SHARP materials used on the cowl leading edge?
<BR> <INPUT TYPE="radio" NAME="sharp" VALUE="yes" CHECKED>Yes
   <INPUT TYPE="radio" NAME="sharp" VALUE="no">No

<! Radio buttons for selecting family of TPS materials>
<P> Select TPS material family for the analysis:
<BR> <INPUT TYPE="radio" NAME="tpsfamily" VALUE="all" CHECKED>All Materials in TCAT
   <INPUT TYPE="radio" NAME="tpsfamily" VALUE="nextgen">Next Gen RLV Materials
   <INPUT TYPE="radio" NAME="tpsfamily" VALUE="shuttle">Shuttle Gen RLV Materials

<BR>
<A HREF="http://asm.arc.nasa.gov/cgi-bin/tpsx/unrestrict/V2/tpsx-frame.pl" TARGET=new>
TPSX Material Properties Data Base </A>

<! Textbox for entering the name of MINIVER file >
<P> Enter name of MINIVER output file:
<BR> <INPUT TYPE="text" NAME="filename" SIZE="30"
    MAXLENGTH="30">

<BR>
<BR>
<! Create the submit button>
<INPUT NAME="Submit" TYPE=submit VALUE="Run TCAT">
<INPUT NAME="Reset"  TYPE=reset VALUE="Reset Inputs">
<BR>
<BR>
<A HREF="tcat_selection.html">Return to selection page</A>
 <HR>
 </FORM>
</BODY>
</HTML>
```

APPENDIX F

TCAT MULTIPLE TPS MATERIAL DESIGN HTML FILES

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Translational//EN"><HTML>
<HEAD>

  <TITLE> TCAT-bodycalcs-inputs </TITLE>

</HEAD>
<BODY>
  <P> You have selected the body TPS design option.
    Please fill in the fields below. </P>
  <HR>
  <FORM METHOD=post ACTION="tcat_bodycalc.cgi" TARGET="TCAT_outputs">

    <P> Enter the approximate wetted body area (ft^2).
    <BR>
    <INPUT TYPE="text" NAME="bodyarea" SIZE="10" MAXLENGTH="10">

    <P> What fraction of body area is covered with tiles?
    <BR>
    <INPUT TYPE="text" NAME="bodytile" SIZE="10" MAXLENGTH="10">

    <P> What fraction of body area is covered with blankets?
    <BR>
    <INPUT TYPE="text" NAME="bodyblanket" SIZE="10" MAXLENGTH="10"> </P>

    <P> Are SHARP materials used on the nose?
    <BR> <INPUT TYPE="radio" NAME="sharp" VALUE="yes" CHECKED>Yes
      <INPUT TYPE="radio" NAME="sharp" VALUE="no">No

    <! Textbox for entering the name of MINIVER file >
    <P> Enter name of MINIVER output file:
    <BR> <INPUT TYPE="text" NAME="filename" SIZE="30"
       MAXLENGTH="30">

    <! Radio buttons for selecting side of vehicle to analyze>
    <P> Select side of vehicle for analysis:
    <BR> <INPUT TYPE="radio" NAME="side" VALUE="windward" CHECKED>Windward
      <INPUT TYPE="radio" NAME="side" VALUE="leeward">Leeward

    <! Menu for TPS selection>
    <P> Select the type of TPS to be used:
    <BR>
    <SELECT NAME="tps" SIZE="5">
      <OPTION>------TILES------
      <OPTION VALUE="AETB8_Al">AETB8_Al_backface
      <OPTION VALUE="AETB8_GrEx">AETB8_GrEx_backface
      <OPTION VALUE="AETB12_Al">AETB12_Al_backface
      <OPTION VALUE="AETB12_GrEx">AETB12_GrEx_backface
      <OPTION VALUE="FRCI12_Al">FRCI12_Al_backface
      <OPTION VALUE="FRCI12_GrEx">FRCI12_GrEx_backface
      <OPTION VALUE="FRCI20_Al">FRCI20_Al_backface
      <OPTION VALUE="FRCI20_GrEx">FRCI20_GrEx_backface
      <OPTION VALUE="LI900_Al">LI900_Al_backface
      <OPTION VALUE="LI900_GrEx">LI900_GrEx_backface
      <OPTION VALUE="LI2200_Al">LI2200_Al_backface
      <OPTION VALUE="LI2200_GrEx">LI2200_GrEx_backface
      <OPTION SELECTED VALUE="RCC_GrEx">RCC_GrEx_backface
      <OPTION VALUE="RCC_Al">RCC_Al_backface
      <OPTION VALUE="SiC_GrEx">SiC_GrEx_backface
      <OPTION VALUE="SiC_Al">SiC_Al_backface
```

```
      <OPTION VALUE="TUFI_GrEx">TUFI_GrEx_backface
      <OPTION VALUE="TUFI_Al">TUFI_Al_backface
      <OPTION>-----BLANKETS-----
      <OPTION VALUE="AFRSI_Al">AFRSI_Al_backface
      <OPTION VALUE="AFRSI_GrEx">AFRSI_GrEx_backface
      <OPTION VALUE="CFBI_Al">CFBI_Al_backface
      <OPTION VALUE="CFBI_GrEx">CFBI_GrEx_backface
      <OPTION VALUE="DURAFRSI_Al">DURAFRSI_Al_backface
      <OPTION VALUE="DURAFRSI_GrEx">DURAFRSI_GrEx_backface
      <OPTION VALUE="DURAFRSI_Al">DURAFRSI_Al_backface
      <OPTION VALUE="DURAFRSI_GrEx">DURAFRSI_GrEx_backface
      <OPTION VALUE="PBI_Al">PBI_Al_backface
      <OPTION VALUE="PBI_GrEx">PBI_GrEx_backface
      <OPTION VALUE="TABI_Al">TABI_Al_backface
      <OPTION VALUE="TABI_GrEx">TABI_GrEx_backface
    </SELECT>
    <BR>
    <A HREF="http://asm.arc.nasa.gov/cgi-bin/tpsx/unrestrict/V2/tpsx-frame.pl">
    TPSX Material Properties Data Base </A>
    <BR>
    <BR>
    <! Create the submit button>
    <INPUT NAME="Submit" TYPE=submit VALUE="Run TCAT">
    <INPUT NAME="Reset"  TYPE=reset VALUE="Reset Inputs">
  <HR>
  </FORM>
</BODY>
</HTML>



<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Translational//EN"><HTML>
<HEAD>

  <TITLE> TCAT-wingcalcs-inputs </TITLE>

</HEAD>
<BODY>
  <P> You have selected the wing and tail TPS design option.
    Please fill in the fields below. </P>
  <HR>
  <FORM METHOD=post ACTION="tcat_wingcalc.cgi" TARGET="TCAT_outputs">

    <P> Enter the approximate wetted wing area (ft^2).
    <BR>
    <INPUT TYPE="text" NAME="wingarea" SIZE="10" MAXLENGTH="10">

    <P> What fraction of wing area is covered with tile TPS?
    <BR>
    <INPUT TYPE="text" NAME="wingtile" SIZE="10" MAXLENGTH="10">

    <P> What fraction of wing area is covered with blanket TPS?
    <BR>
    <INPUT TYPE="text" NAME="wingblanket" SIZE="10" MAXLENGTH="10"> </P>


    <P> Are SHARP materials used on the wing leading edges?
    <BR> <INPUT TYPE="radio" NAME="sharp" VALUE="yes" CHECKED>Yes
```

```
    <INPUT TYPE="radio" NAME="sharp" VALUE="no">No

<P> If yes, enter the length of the exposed wing leading edge(ft).
<BR>
<INPUT TYPE="text" NAME="leadingedge" SIZE="10" MAXLENGTH="10">

<! Textbox for entering the name of MINIVER file >
<P> Enter name of MINIVER output file:
<BR> <INPUT TYPE="text" NAME ="filename" SIZE="30"
     MAXLENGTH="30">

<! Radio buttons for selecting side of vehicle to analyze>
<P> Select side of vehicle for analysis:
<BR> <INPUT TYPE="radio" NAME="side" VALUE="windward" CHECKED>Windward
    <INPUT TYPE="radio" NAME="side" VALUE="leeward">Leeward

<! Menu for TPS selection>
<P> Select the type of TPS to be used:
<BR>
<SELECT NAME="tps" SIZE="5">
  <OPTION>------TILES------
  <OPTION VALUE="AETB8_Al">AETB8_Al_backface
  <OPTION VALUE="AETB8_GrEx">AETB8_GrEx_backface
  <OPTION VALUE="AETB12_Al">AETB12_Al_backface
  <OPTION VALUE="AETB12_GrEx">AETB12_GrEx_backface
  <OPTION VALUE="FRCI12_Al">FRCI12_Al_backface
  <OPTION VALUE="FRCI12_GrEx">FRCI12_GrEx_backface
  <OPTION VALUE="FRCI20_Al">FRCI20_Al_backface
  <OPTION VALUE="FRCI20_GrEx">FRCI20_GrEx_backface
  <OPTION VALUE="LI900_Al">LI900_Al_backface
  <OPTION VALUE="LI900_GrEx">LI900_GrEx_backface
  <OPTION VALUE="LI2200_Al">LI2200_Al_backface
  <OPTION VALUE="LI2200_GrEx">LI2200_GrEx_backface
  <OPTION VALUE="RCC_GrEx">RCC_GrEx_backface
  <OPTION VALUE="RCC_Al">RCC_Al_backface
  <OPTION VALUE="SiC_GrEx">SiC_GrEx_backface
  <OPTION VALUE="SiC_Al">SiC_Al_backface
  <OPTION SELECTED VALUE="TUFI_GrEx">TUFI_GrEx_backface
  <OPTION VALUE="TUFI_Al">TUFI_Al_backface
  <OPTION>-----BLANKETS-----
  <OPTION VALUE="AFRSI_Al">AFRSI_Al_backface
  <OPTION VALUE="AFRSI_GrEx">AFRSI_GrEx_backface
  <OPTION VALUE="CFBI_Al">CFBI_Al_backface
  <OPTION VALUE="CFBI_GrEx">CFBI_GrEx_backface
  <OPTION VALUE="DURAFRSI_Al">DURAFRSI_Al_backface
  <OPTION VALUE="DURAFRSI_GrEx">DURAFRSI_GrEx_backface
  <OPTION VALUE="PBI_Al">PBI_Al_backface
  <OPTION VALUE="PBI_GrEx">PBI_GrEx_backface
  <OPTION VALUE="TABI_Al">TABI_Al_backface
  <OPTION VALUE="TABI_GrEx">TABI_GrEx_backface
</SELECT>
<BR>
<A HREF="http://asm.arc.nasa.gov/cgi-bin/tpsx/unrestrict/V2/tpsx-frame.pl">
TPSX Material Properties Data Base</A>
<BR>
<BR>
<! Create the submit button>
<INPUT NAME="Submit" TYPE=submit VALUE="Run TCAT">
<INPUT NAME="Reset"  TYPE=reset VALUE="Reset Inputs">
```

```
      <BR>
      <BR>
      <A HREF="tcat_selection.html">Return to selection page</A>
    <HR>
    </FORM>
</BODY>
</HTML>



<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Translational//EN"><HTML>
<HEAD>

  <TITLE> TCAT-cowlcalcs-inputs</TITLE>

</HEAD>
<BODY>
  <P> You have selected the cowl TPS design option.
     Please fill in the fields below.</P>
  <HR>
  <FORM METHOD=post ACTION="tcat_cowlcalc.cgi" TARGET="TCAT_outputs">

    <P> Enter the approximate wetted cowl area (ft^3).
    <BR>
    <INPUT TYPE="text" NAME="cowlarea" SIZE="10" MAXLENGTH="10">

    <P> What fraction of cowl area is covered with tile TPS?
    <BR>
    <INPUT TYPE="text" NAME="cowltile" SIZE="10" MAXLENGTH="10">

    <P> What fraction of cowl area is covered with blanket TPS?
    <BR>
    <INPUT TYPE="text" NAME="cowlblanket" SIZE="10" MAXLENGTH="10"> </P>

    <P> Are SHARP materials used on the cowl leading edges?
    <BR> <INPUT TYPE="radio" NAME="sharp" VALUE="yes" CHECKED>Yes
        <INPUT TYPE="radio" NAME="sharp" VALUE="no">No

    <P> If yes, enter the length of the exposed cowl leading edge(ft).
    <BR>
    <INPUT TYPE="text" NAME="leadingedge" SIZE="10" MAXLENGTH="10">

    <! Textbox for entering the name of MINIVER file >
    <P> Enter name of MINIVER output file:
    <BR> <INPUT TYPE="text" NAME="filename" SIZE="30"
         MAXLENGTH="30">

    <! Radio buttons for selecting side of vehicle to analyze>
    <P> Select side of vehicle for analysis:
    <BR> <INPUT TYPE="radio" NAME="side" VALUE="windward" CHECKED>Windward
        <INPUT TYPE="radio" NAME="side" VALUE="leeward">Leeward

    <! Menu for TPS selection>
    <P> Select the type of TPS to be used:
    <BR>
    <SELECT NAME="tps" SIZE="5">
      <OPTION>------TILES------
      <OPTION VALUE="AETB8_Al">AETB8_Al_backface
```

```
        <OPTION VALUE="AETB8_GrEx">AETB8_GrEx_backface
        <OPTION VALUE="AETB12_Al">AETB12_Al_backface
        <OPTION VALUE="AETB12_GrEx">AETB12_GrEx_backface
        <OPTION VALUE="FRCI12_Al">FRCI12_Al_backface
        <OPTION VALUE="FRCI12_GrEx">FRCI12_GrEx_backface
        <OPTION VALUE="FRCI20_Al">FRCI20_Al_backface
        <OPTION VALUE="FRCI20_GrEx">FRCI20_GrEx_backface
        <OPTION VALUE="LI900_Al">LI900_Al_backface
        <OPTION VALUE="LI900_GrEx">LI900_GrEx_backface
        <OPTION VALUE="LI2200_Al">LI2200_Al_backface
        <OPTION VALUE="LI2200_GrEx">LI2200_GrEx_backface
        <OPTION VALUE="RCC_GrEx">RCC_GrEx_backface
        <OPTION VALUE="RCC_Al">RCC_Al_backface
        <OPTION VALUE="SiC_GrEx">SiC_GrEx_backface
        <OPTION VALUE="SiC_Al">SiC_Al_backface
        <OPTION SELECTED VALUE="TUFI_GrEx">TUFI_GrEx_backface
        <OPTION VALUE="TUFI_Al">TUFI_Al_backface
        <OPTION>-----BLANKETS-----
        <OPTION VALUE="AFRSI_Al">AFRSI_Al_backface
        <OPTION VALUE="AFRSI_GrEx">AFRSI_GrEx_backface
        <OPTION VALUE="CFBI_Al">CFBI_Al_backface
        <OPTION VALUE="CFBI_GrEx">CFBI_GrEx_backface
        <OPTION VALUE="DURAFRSI_Al">DURAFRSI_Al_backface
        <OPTION VALUE="DURAFRSI_GrEx">DURAFRSI_GrEx_backface
        <OPTION VALUE="DURAFRSI_Al">DURAFRSI_Al_backface
        <OPTION VALUE="DURAFRSI_GrEx">DURAFRSI_GrEx_backface
        <OPTION VALUE="PBI_Al">PBI_Al_backface
        <OPTION VALUE="PBI_GrEx">PBI_GrEx_backface
        <OPTION VALUE="TABI_Al">TABI_Al_backface
        <OPTION VALUE="TABI_GrEx">TABI_GrEx_backface
      </SELECT>
      <BR>
      <A HREF="http://asm.arc.nasa.gov/cgi-bin/tpsx/unrestrict/V2/tpsx-frame.pl">
      TPSX Material Properties Data Base</A>
      <BR>
      <BR>
      <! Create the submit button>
      <INPUT NAME="Submit" TYPE=submit VALUE="Run TCAT">
      <INPUT NAME="Reset"  TYPE=reset VALUE="Reset Inputs">
      <BR>
      <BR>
      <A HREF="tcat_selection.html">Return to selection page</A>
    <HR>
    </FORM>
  </BODY>
</HTML>
```

REFERENCES

1. Engel, C.D. and Konishi, S., "MINIVER Upgrade for the AVID System", NASA CR 172213, August 1993.

2. NASA Ames Thermal Protection Materials and System Branch, TPS*X* Database Internet Site: http://asm.arc.nasa.gov.

3. Cowart, K., and Olds, J, "Integrating Aeroheating and TPS Into Conceptual RLV Design," AIAA 99-4806, November 1999.

4. Vanderplaats, G.N., "ADS – A Fortran Program For Automated Design Synthesis – Version 1.10," NASA-CR 177985, September, 1985.

5. Olds, J.R., "Launch Vehicle Systems Analysis," NAG8-1302, January 1999.

6. Olds, J., Ledsinger, L., "*Stargazer:* A TSTO Bantam-X Vehicle Concept Utilizing Rocket Based Combined-Cycle Propulsion," AIAA 99-4888, November 1999.

7. Macconochie, I.O., and Klich, P.J., "Techniques For The Determination of Mass Properties of Earth-To-Orbit Transportation Systems," NASA TM 78661, June 1978.

8. Engel, C.D., and Schmitz, C.P., "Upgrade For the AVID System, Volume II: LANMIN Input Guide," NAS1-17896, Remtech, Inc., Alabama, May 1987.

9. Behee, R., "Introduction to Thermal Modeling Using SINDA/G, A Tutorial Guide," Network Analysis Associates, Inc., California, 9th Edition, June 1994.

10. Holzner, S., Perl Core Language: Little Black Book, Coriolis Technology Press, 1999.

11. Norman, I., Rochelle, W.C, Kimbrough, B.S., "Comparison of Orbiter STS-2 Development Flight Instrumentation Data with Thermal Math Model Predictions," AIAA 82-0839, June 1982.

12. Dotts, R.L., Tillian, D.J., and Smith, J.A., "Space Shuttle Orbiter-Reusable Surface Insulation Flight Performance," Paper 82-0788CP, AIAA/ASME/ASCE/AHS 23rd Structural Dynamics, and Materials Conference, Orlando, Fla. May 1982.

13. Rochelle, W.C., Battley, H.H., and Gallegos, J.J., "Use of Arc-Jet Hypersonic Blunted Wedge Flows for Evaluating Performance of Orbiter TPS," Entry Heating and Thermal Protection: AIAA Progress in Aeronautics and Astronautics, Vol. 69, edited by W.B. Olstad, AIAA, New York, 1980, pp 287-315.

14. Rochelle, W.C., Battley, H.H., Hale, W.M., Gallegos, J.J., and Kimbrough, B.S., "Arc-Jet Test and Analysis of Orbiter TPS Intertile Heating in High Pressure Gradient Flow," Paper 78-845, AIAA/ASME 2nd Thermophysics and Heat Transfer Conference, Palo Alto, Calif., May 1978.

15. Chen, Y.K., Milos, F.S, "Solution Strategy for Thermal Response of Nonablating Thermal Protection System Materials at Hypersonic Speeds," AIAA 96-0615.

16. Access to Space Study Team, Volume III: Final Report – Design Databook, July, 1993.

17. Lashkarl, M., COSMOS/M User's Guide, Volume 1, Structural Research and Analysis Corp., Santa Monica, Calif., April 1992.

18. McGrory, D.M., Slack, D.C., Applebaum, M.P. and Walters, R.W., GASP Version 2.2 User's Manual, Aerosoft, Inc., 1993.

19. Olynick, D., "Trajectory-Based Thermal Protection System Sizing for an X-33 winged Vehicle Concept," *Journal of Spacecraft and Rockets*, Volume 33, No.3, May-June 1998.

20. Brauer, G. L., Cornick, D.E., and Stevenson, R., "Capabilites and Applications of the Program to Optimize Simulated Trajectories," NASA-CR 2770, Feb, 1977.

21. Press, W.H., Teukolsky, S.A., Vetterling,W.T., Flannery, B.P., Numerical Recipes in Fortran 7, 2nd Edition, Cambridge University Press, 1992.